Ensembles of Neural Networks for Tumour Motion Prediction

by

Neil Wallace Johnson

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Medical Physics

Department of Oncology

University of Alberta

Abstract

Dynamic tumour-tracked radiotherapy is a promising method for delivering conformal doses to tumours that exhibit a large degree of motion as a result of patient respiration. However, there exists an inevitable latency between the acquisition of an image of a moving tumour and the adaptation of the therapeutic beam to match its observed position and contour. This must be addressed by predicting respiration-induced tumour motion so that the requisite mechanical adjustments can be initiated sufficiently in advance. For MR-based tracking, this latency is relatively long compared to other imaging techniques. Accurate motion prediction therefore requires a more sophisticated approach than those used for short-latency hardware.

A novel application of long short-term memory recurrent neural networks for respiration-induced tumour motion is presented in this thesis. It consists of three main components: (1) acceleration of training using super-convergence regularization with intelligent early stopping; (2) mitigation of overfitting and instability through homogeneous network ensembles; and (3) improvement of the reaction to changing respiratory patterns during treatment through a novel adaptation method called intermittent retraining. Compared to previous studies, this approach reduces the amount of time required for network training by several orders of magnitude while simultaneously improving the accuracy and consistency of predictions. This work represents a step toward bringing linac-MR based dynamic tumour-tracked radiotherapy into clinical relevance by making it both more practical and more precise.

This work is dedicated to Sarah, who supported and encouraged me throughout my courses, research and the writing of this thesis, and to Teddy and Maddie, who are constant sources of joy and inspiration in my life.

It is also dedicated to the memory of my father-in-law Rocke, and to everyone else that cancer has taken from us too soon.

Acknowledgements

First and foremost, I would like to acknowledge the guidance, support, patience and expertise offered by my supervisor, Dr. Jihyun Yun, throughout my research and the process of writing this thesis. I would also like to thank my supervisory committee members, Dr. Gino Fallone, Dr. Satyapal Rathee and Dr. Keith Wachowicz for their time, attention, excellent questions and constructive input. Thank you as well to Dr. Brad Warkentin for his time and effort in joining the thesis defence committee, and to Dr. Hans-Sonke Jans for serving as chair.

The faculty and staff at the Cross Cancer institute have provided me with countless opportunities to learn, work and grow as a person while pursuing my graduate degree, and I will always be grateful for the experience. While the pandemic has prevented me from physically being with my colleagues and fellow students as much as I would have liked, my memories of this time are still overwhelmingly positive.

Contents

List of Tables

List of Figures

List of Symbols

$\sim$	Approximately
$\bar{A}$	Mean Amplitude
$\vec{B}_0$	Main MR Magnetic Field
$\vec{B}_1(t)$	Pulsed Excitation Magnetic Field
c	Cell State (LSTM)
C	Cost Function
$\bar{C}$	Mean Cost Function
$\widetilde{C}$	Amplitude-Normalized Mean Cost Function
$\vec{G}$	Gradient Magnetic Field
h	Hidden State
$\hbar$	Reduced Planck Constant
^1H	Hydrogen Nucleus
k_B	Boltzmann Constant
$\vec{M}_0$	Net Magnetization Vector
N_{ens}	Number of Networks in Ensemble
N_{HL}	Number of Hidden Layers
N_{seq}	Length of Input Sequence
$\vec{S}$	Nuclear Spin
t_{acq}	Acquisition Time
t_{sys}	System Delay
T_R	Repetition Time
$\mathbf{U}, \mathbf{V}, \mathbf{W}$	Weight Matrices
w_{HL}	Width of Hidden Layer(s)
$\vec{X}$	Input Vector
$\vec{Y}$	Output Vector
$\hat{Y}$	Ground-Truth Output
$\vec{Z}$	Weighted Summed Input
α	Learning Rate (Section 2.1)
α	Nutation Angle (Section 2.2)
α_i	Initial Learning Rate
γ	Gyromagnetic Ratio
$\vec{\mu}$	Nuclear Magnetic Moment
$\vec{\tau}$	Torque

ϕ	Activation Function
ω_0	Larmor Frequency

List of Abbreviations

2D	Two-Dimensional
3D	Three-Dimensional
A/P	Anterior/Posterior
ABC	Active Breathing Control
ANN	Artificial Neural Network
bSSFP	Balanced Steady-State Free Precession
CBCT	Cone-Beam Computed Tomography
CNR	Contrast-to-Noise Ratio
CT	Computed Tomography
CTV	Clinical Target Volume
DIBH	Deep Inspiration Breath-Hold
DMLC	Dynamic Multi-Leaf Collimator
DTTRT	Dynamic Tumour-Tracked Radiotherapy
EBRT	External Beam Radiation Therapy
EPI	Echo-Planar Imaging
EPID	Electronic Portal Imaging Device
FSB	Forced Shallow Breathing
GPU	Graphics Processing Unit
GTV	Gross Tumour Volume
HPO	Hyperparameter Optimization
ICRU	International Commission on Radiological Units and Measurements
IGRT	Image-Guided Radiotherapy
IM	Internal Margin
IMRT	Intensity-Modulated Radiotherapy
IR	Intermittent Retraining
ITV	Internal Target Volume
kV	Kilovoltage
L/R	Left/Right
linac	Linear Accelerator
LR	Learning Rate
LReLU	Leaky Rectified Linear Unit
LSTM	Long Short-Term Memory
MR	Magnetic Resonance
MRI	Magnetic Resonance Imaging
MV	Megavoltage

nifteRT	Non-Invasive Intrafractional Tumour-Tracked Radiotherapy
NN	Neural Network
PTV	Planning Target Volume
ReLU	Rectified Linear Unit
RF	Radiofrequency
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
S/I	Superior/Inferior
SM	Setup Margin
SSFP	Steady-State Free Precession
tanh	Hyperbolic Tangent
TSE	Turbo Spin Echo
US	Ultrasound

Chapter 1

Introduction

1.1 Thesis Organization

Generally, this thesis describes a novel strategy for predicting respiration-induced tumour motion during external beam radiation therapy (EBRT) using recurrent neural networks (RNNs). In Chapter 1, the role of tumour motion prediction in dynamic tumour-tracked radiotherapy (DTTRT) is explained, and the need for non-linear prediction when performing DTTRT on a hybrid linear accelerator (linac) and magnetic resonance (MR) imaging system is identified. In Chapter 2, the basic theory behind artificial neural networks (ANNs) and RNNs is described, with a specific focus on modifications to the standard ANN approach that are required when training data are limited. The fundamental concepts of magnetic resonance imaging (MRI) are then introduced, with a special focus on the determinants of MRI acquisition speed. Chapter 3 contains a manuscript that presents the specific details of the novel approach to RNN training that has been developed, and quantifies the effects of this approach on training time and predictive accuracy. A version of this chapter has been submitted to *Medical Physics* for publication and, as of this writing, is undergoing peer-review. Finally, Chapter 4 revisits the most salient points of this thesis and suggests potential future avenues of research related to this project.

1.2 Cancer Incidence and Treatment Strategies

Cancer is currently the leading cause of death in Canada, and it is estimated that more than 40% of Canadians will be diagnosed with cancer during their lifetimes [1]. Lung cancer is currently the most prevalent form, estimated to represent 13% of new cancer diagnoses and 25% of cancer-related deaths in Canada in 2022 [1].

Treatment strategies for cancer vary considerably depending on the primary disease site, the genetics of the cancer cells, the stage of the disease at diagnosis and the preferences and age of the patient. Treatments may be primary, adjuvant or palliative, depending on whether they are intended to be curative, mitigate the risk of recurrence, or alleviate symptoms of disease and prolong survival, respectively. Common therapeutic approaches include surgery, pharmacotherapy (cytotoxic chemotherapy, hormone therapy, targeted therapy), radiation therapy (EBRT, brachytherapy, radioisotope therapy), and immunotherapy. In most cases, a patient's full treatment course will involve a combination of these strategies.

1.3 External Beam Radiation Therapy

EBRT is the most widely-used radiation therapy modality [2]. It is estimated that about half of all cancer patients receive EBRT throughout their treatment, and that about a quarter will require more than one course [3]. EBRT broadly refers to a variety of treatments in which ionizing radiation is delivered to the cancer cells in the form of a beam generated outside of the patient's body. Most commonly, the radiation takes the form of megavoltage (MV) X-rays or electrons produced by a medical linac, but kilovoltage (kV) photons from an X-ray tube (orthovoltage radiation therapy), gamma rays emitted by an

external radioactive source (as in cobalt-60 devices) or charged particles that have been energized in a cyclotron (as in proton or carbon ion therapy) are also possible. This thesis exclusively considers MV photon EBRT delivered via linac, more specifically, a hybrid linac-MR.

1.4 Conformality, Volumes, and Margins in EBRT

A long-standing goal in the field of radiation therapy is to maximize the *conformality* of the treatment, which means escalating the ionizing radiation dose to tissues that either contain visible cancer or are likely to host subclinical disease while simultaneously reducing the dose to surrounding healthy tissues. It is a generally accepted notion that any improvements to conformality will both increase the probability of local tumour control[4]–[6] and reduce the probability of adverse effects[7], [8], though the clinical evidence supporting each iteration of improved conformality inevitably lags behind its implementation due to the need for long-term follow-up in cancer studies.

There are many potential uncertainties in prescribing, planning and delivering EBRT that limit the achievable treatment conformality. These must be well-understood and properly accounted for – attempting to deliver an unrealistically conformal treatment can result in underdosing a portion of the tumour, increasing the likelihood of disease progression or recurrence. To address these uncertainties, a standardized approach to prescribing and reporting doses in EBRT has been developed by the International Commission on Radiological Units and Measurements (ICRU). It is laid out in ICRU Report 50 [9], updated to address advances in EBRT in its supplement, ICRU Report 62 [10], and updated again to account for the high dose gradients associated with intensity-modulated radiation therapy (IMRT) in ICRU Report 83 [11].

These reports recommend identifying four volumes of interest prior to treat-

ment, as indicated in Figure 1.1. The gross tumour volume (GTV) encompasses the observable extent of the cancerous tissue. To account for the likely invasion of subclinical disease in the tissues around the bulk tumour, this volume is expanded at the discretion of the radiation oncologist to generate the clinical target volume (CTV). It is to the CTV that the prescribed dose is intended to be delivered, and outside of it that the dose should be minimized.

However, this task is complicated by the fact that the CTV is not static in terms of its size, shape nor position relative to radiation isocenter. Motion of the CTV can take many forms – it can be either interfractional (taking place between subsequent treatment fractions) or intrafractional (taking place during a single treatment fraction), and it can be either internal (the CTV moving with respect to the patient frame) or external (the patient frame moving with respect to radiation isocenter).

In order to address internal motion, ICRU Report 62 recommends the addition of an internal margin (IM) that encompasses the anticipated internal range of the CTV during treatment. Together, the CTV and IM yield a new volume called the internal target volume (ITV). This ITV is then further expanded by a setup margin (SM) to account for potential uncertainties in aligning the patient frame to radiation isocenter. Together, the CTV, IM and SM finally yield the planning target volume (PTV). By delivering the prescribed radiation dose to a properly representative PTV, it can be ensured that the CTV will receive complete dose coverage.

In practice, achieving the best therapeutic ratio (the tradeoff between tumour control probability and normal tissue complication probability) means that the margins should be made as small as possible, so that the volume of healthy tissue receiving the prescribed radiation dose (PTV - CTV) is minimized. Doing so without compromising the probability of a successful treatment requires a thorough understanding of the potential sources of target

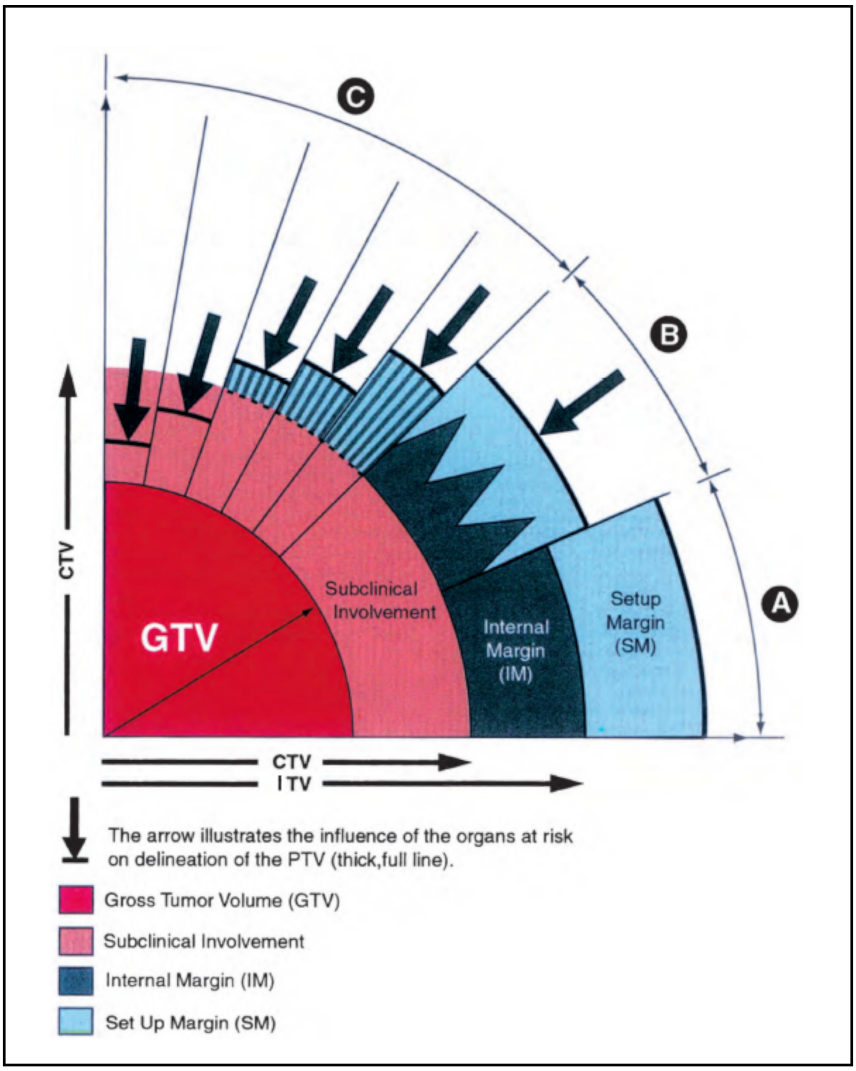

Figure 1.1: Treatment volumes and scenarios as defined by ICRU Report 62[10]. Reproduced with permission from SAGE Publishing, see Appendix A. The major treatment volumes (the GTV, CTV, ITV and PTV) are identified. Three cases for margin addition are illustrated: (A) linear addition, which provides the best dose coverage of the CTV but results in the highest normal tissue dose, (B) a stastistically rigorous addition, which is possible when the sources of motion are well-characterized, and (C) smaller margins than warranted by the motion in order to spare normal tissues.

motion.

From a statistical standpoint, sources of motion can be characterized by their mean magnitude Σ and their standard deviation σ. In the parlance of statistics, these values represent the systematic and random errors of the

tumour position, respectively. If all sources of motion were identified and characterized prior to treatment, a simple method to assign margins would be to simply sum all of the potential errors together linearly. However, this would often result in very large margins, reducing the conformality of the treatment and potentially intersecting nearby organs at risk. A statistical approach has been outlined by Stroom[12] who determined the margin M_S required to deliver a minimum of 95% of the prescription dose to an average of 99% of the CTV:

$$M_S = 2\Sigma + 0.7\sigma \tag{1.1}$$

Van Herk[13] followed up shortly thereafter with a formula that yields a slightly larger margin M_{VH}, such that 90% of patients will receive a minimum dose of 95% the prescription dose to their CTV:

$$M_{VH} = 2.5\Sigma + 0.7\sigma \tag{1.2}$$

However, these more rigorously determined margins may still be too large in the case of high-amplitude intrafractional motion or close proximity to organs at risk. In Figure 1.1, the ICRU 62 report illustrates three potential scenarios for margin addition, depending on constraints regarding healthy tissue dose limits. In scenario A, dose outside of the CTV is of relatively little concern, so the IM and SM are added linearly to the CTV to produce the PTV, ensuring full dose coverage. In scenario B, the uncertainties that underpin the IM and SM are well understood and a minimization of dose outside the CTV is desired, so they are added together in a more rigorous statistical manner. In scenario C, in order to reduce the dose to nearby organs at risk and minimize the risk of intolerable complications, smaller margins must be used and the CTV will not get full dose coverage. This will result in a lower probability of tumour control. In the event that the GTV cannot safely be covered by the prescribed

dose, the treatment becomes palliative.

1.5 Interfractional Motion

1.5.1 External Interfractional Motion

The major contributor to external interfractional motion is variability in patient setup. In the absence of image guidance, which will be discussed in Section 1.6, the patient is conventionally positioned by aligning external skin markers to room lasers that are meant to indicate radiation isocenter. This carries with it uncertainties associated with the coincidence of laser and radiation isocenter, the potential for marker motion relative to internal anatomy, and variability in setup on different days and between different radiation therapists. Patient immobilization devices can help ensure a more consistent setup[14].

1.5.2 Internal Interfractional Motion

One source of internal interfractional motion can be the result of day-to-day differences in bowel or bladder filling. If not controlled, this can cause the CTV to move up to several cm [15], depending on the proximity of the tumour site to the bladder and digestive tract. Consistent patient preparation prior to simulation and treatment can help to mitigate this variation[16].

Additionally, throughout a course of EBRT there can be considerable changes in patient anatomy (e.g., weight loss or gain) and the volume and shape of the tumour (e.g, swelling, progression or reduction). These types of internal motion are more complicated to address, often requiring replanning of the treatment rather than simple adjustment of the patient. This is a motivation for adaptive radiotherapy[17], which is beyond the scope of this thesis.

1.6 Image-Guided EBRT for Interfractional Motion

An improvement on the laser-based setup approach is to directly image the patient at the beginning of a treatment fraction, allowing for setup to be based on rigid anatomical features that are more proximal to the tumour (or, in some cases, the tumour itself). This process is known as image-guided radiotherapy (IGRT)[18]. With IGRT, the required SM is much smaller since it only needs to reflect variability in aligning the pre-treatment images to simulation images (as well any potential disagreement between imaging and radiation isocenter). It can be performed with a range of imaging modalities, each having their own strengths and weaknesses.

Most modern linacs are equipped with a flat panel detector called an electronic portal imaging device (EPID) that can be positioned along the path of the therapeutic beam, as well as a kV imaging system located orthogonal to the therapeutic beam. Both can be used to rapidly acquire two-dimensional (2D) setup verification images at the beginning of (or throughout) treatment. kV imaging provides better contrast-to-noise ratios (CNRs) because attenuation differences are larger between different tissues in the kV spectrum. This results in better image quality at lower patient doses. MV imaging, however, does come with the added benefits of guaranteed coincidence between imaging and radiation isocenter and the suppression of image artefacts caused by scattering from high atomic number materials such as hip prostheses.

The quality of setup matching that can be achieved from planar imaging is limited, since information about the patient's three-dimensional (3D) positioning is inevitably lost when their anatomy is represented in 2D. In MV- and kV-based cone-beam computed tomography (CBCT), multiple 2D projections are acquired as the linac rotates, yielding a volumetric representation of

the patient when they are backprojected. The improved setup matching provided by CBCT comes at the cost of increased patient dose relative to planar imaging, as well as increased acquisition time. As a result, CBCT imaging is only viable at the beginning of treatment fractions, which is adequate for addressing interfractional motion.

There are some non-ionizing imaging modalities for IGRT for which patient dose is not a concern. Ultrasound (US) imaging uses high-frequency acoustic waves to detect variation in acoustic impedance within the patient. It can often provide adequate soft tissue contrast to visualize tumours directly, and when paired with methods to localize the US probe within the treatment room, has been used to detect both internal and external interfractional motion [19]. However, there is a considerable amount of inter-operator variability for manually obtained US images[20], and therefore variability in setup when using US guidance[21]. Some robotic US probes are currently under development, both to reduce operator dependence and to allow for images to be acquired from outside of the treatment vault, which would enable US-based intrafractional motion compensation as well [22].

Finally, MRI (see Section 2.2 for technical details) is another non-ionizing imaging modality that offers excellent soft-tissue contrast and spatial resolution, often allowing for the direct visualization of tumours. Early MR-IGRT used an MR-on-rails system to translate patients between MR and linac isocenters after validating the setup and adapting for interfractional tumour motion[23]–[25]. Recently, three hybrid linac-MR devices have been developed and clinically deployed and are being used for interfractional motion management [26]–[28]. A specific example of a hybrid Linac-MR (the Alberta Linac-MR[26], [29], the first of its kind) will be discussed in more detail in Section 1.9.

1.7 Intrafractional Motion

1.7.1 External Intrafractional Motion

Even if a perfect alignment between the target and the therapeutic beam is initially achieved, it is unlikely to be maintained throughout treatment. Externally, this is mostly due to the fact that the patient may change their body posture or shift relative to the patient support system, resulting in a displacement from radiation isocenter.

Additionally, any required rotation of the patient support system during setup or treatment might itself result in unintended translations of the patient, and any rotation of the linac itself may shift the position of radiation isocenter relative to imaging isocenter. In modern linacs which are capable of tight tolerances on isocentricity, these potential errors are generally small and can be minimized further with a robust quality assurance program.

1.7.2 Internal Intrafractional Motion

Typically, the predominant source of intrafractional motion is internal and the result of involuntary physiological processes, such as peristalsis, respiration and circulation. The relative contributions of these different processes to intrafractional motion depends on the location of the tumour, but the lungs, chest wall, esophagus, liver, pancreas, breast, prostate and kidneys are all known to move during respiration[30].

This respiration-induced tumour motion is complex. Its magnitude, frequency and regularity can vary widely from patient to patient, from day to day, and even from respiratory cycle to respiratory cycle. Lung tumour motion can be especially complicated (see Figure 1.2), often exhibiting hysteresis (different trajectories between inhalation and exhalation) as well as high-amplitude motion in all three cardinal directions (superior/inferior (S/I), anterior/posterior

(A/P) and left/right (L/R)). Abdominal tumours usually follow more linear paths, most often in the S/I direction, but their motion amplitude can be large as well.

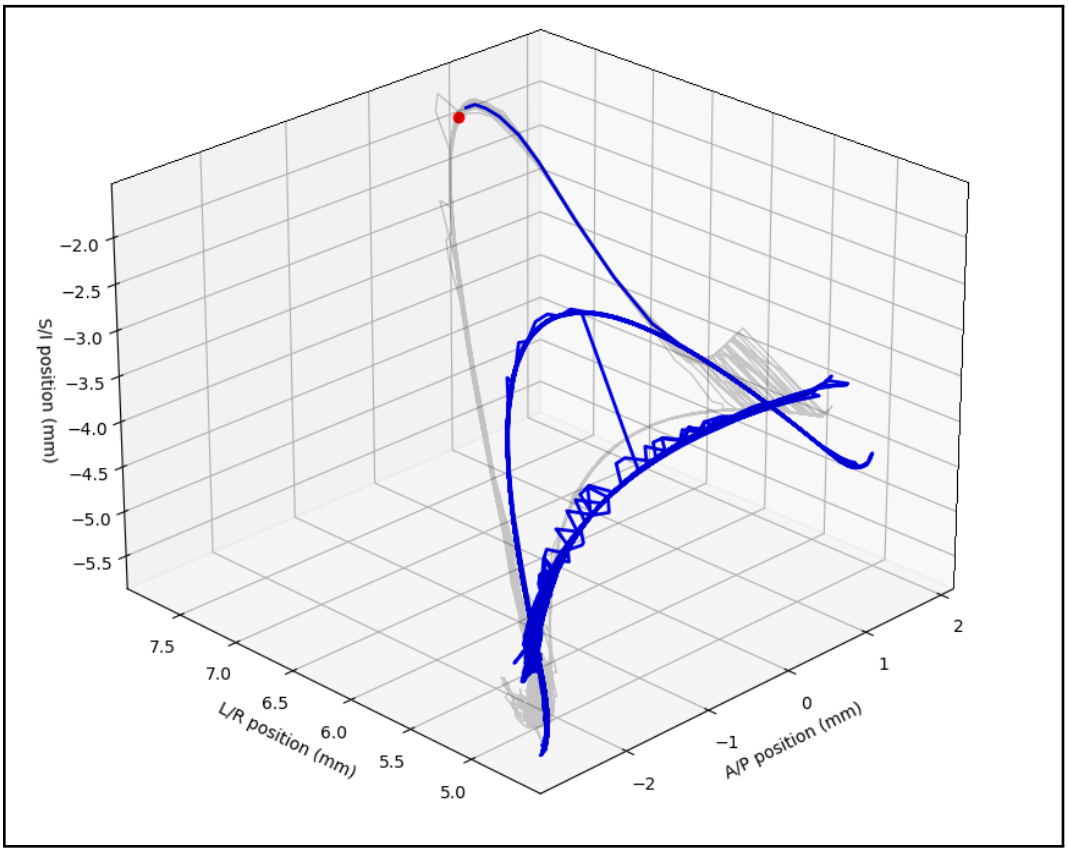

Figure 1.2: A 3D rendering demonstrating the complicated trajectory of a right upper lobe lung tumour, taken from the Suh *et al.* database detailed in Chapter 3. Three distinct trajectories are visible over its entire motion history, and this particular tumour shows a high degree of hysteresis, often switching between the upper and middle paths from inhale to exhale.

Respiratory tumour motion has been studied intensively, owing simultaneously to the prevalence of abdominothoracic tumours, the large proportion of those tumours that have indications for EBRT and the common and often severe consequences of excessive dose to nearby healthy tissue. The lung is particularly sensitive to ionizing radiation, and it is estimated that radiation-induced pneumonitis and lung fibrosis will occur in up to 20% of esophageal and lung cancer patients receiving EBRT[31]. These lung injuries can result

in chronic dyspnea and severely reduced quality of life following treatment.

1.8 Accounting for Intrafractional Motion in EBRT

1.8.1 Traditional Methods

Common clinical approaches to managing respiration-induced intrafractional tumour motion center around either restricting the position of the tumour while the therapeutic beam is active, or activating and deactivating the therapeutic beam based on the detected respiratory phase.

In deep inspiration breath-hold (DIBH)[32], patients are asked to maintain a full inhale state during computed tomography (CT) simulation as well as treatment. While in this state, the position of the tumour will be much more constrained compared to free-breathing, resulting in a smaller requisite IM. Additionally, the tumour often moves away from critical structures such as the heart at maximum inhale. Active breathing control (ABC)[33] is a similar approach, except that the patient is forced to maintain a specific respiratory phase by an external valve rather than doing so voluntarily. In the same vein, forced shallow breathing (FSB)[34] techniques restrict respiratory range through external compression of the abdomen, limiting the motion of the diaphragm and therefore the tumour.

Gated EBRT approaches[35] are based on four-dimensional CT simulation wherein multiple volumetric CT images are obtained, each corresponding to a different respiratory phase. Treatment is restricted to a range of respiratory phases during which the tumour is relatively stable, though some motion (called residual motion) will still occur within this window. Respiratory phase can be measured externally by placing a bellows around the patient's chest to measure thoracic circumference or by placing markers on the patient's skin to

measure external abdominal or thoracic motion, and the detected respiratory phase can be used to automatically trigger beam on/off signals.

All of these approaches are limited by the facts that respiratory phase and tumour position do not have a one-to-one correspondence[36] and that any established correlation is likely to decay over time (e.g., between simulation and treatment). Further, patients are not always able to perform DIBH reproducibly, nor can they always tolerate ABC or FSB[30]. For gated techniques, there needs to be a compromise between the tolerable extent of residual motion and the length of treatments, since treating over too small a respiratory phase range results in a low duty cycle (the ratio of "beam on" time to total treatment time).

1.8.2 Dynamic Tumour-Tracked EBRT

DTTRT techniques involve continual intrafractional measurement of the tumour's position, followed by adaptation of the the therapeutic beam to compensate for any observed motion. Ideally, such an approach would entirely obviate the need for both the IM and the setup component of the SM, since the therapeutic beam is being directly aligned to the tumour (though the potential disagreement between imaging and radiation isocentres would still exist). In practice, however, there are several newly introduced uncertainties that must be accounted for, including errors in tumour localization and accurately calculating and performing the required motion of the beam-steering hardware.

Additionally, there is an inevitable latency between determining the location of the tumour and the completion of the compensatory mechanical motion. This is known as the *system delay*, and if it is not accounted for, it will result in spatial lag of the beam behind the tumour along its direction of motion. As a result, devices that perform DTTRT require some method of

predicting tumour motion associated with respiration, which will be discussed further in Chapter 2. Still, dynamic tumour-tracking has the potential to improve EBRT conformality for highly mobile tumours, and therefore potentially improve outcomes in abdominothoracic cancers.

There have been multiple approaches to the physical adaptation of the therapeutic beam during DTTRT, including using dynamic multi-leaf collimators (DMLCs) [37]–[41], compact gimbals-mounted[42] or robotic[43] linacs, and dynamic patient support systems [44]. Similarly, various tumour sensing approaches have been studied, including fluoroscopic imaging of implanted radiopaque fiducial markers on the periphery of the tumour [45], building and updating a correlation model between external respiratory markers and planar X-rays of internal radiopaque fiducials [46], and detecting implanted radiofrequency (RF) emitting markers [47]. There are several disadvantages to these marker-based approaches: fiducials are known to migrate relative to the tumour, cannot provide a full description of the shape of the GTV, and require an invasive procedure to implant. For thoracic tumours, fiducial marker insertion also carries a pronounced risk of pneumothorax[48].

1.9 Dynamic Tumour Tracking using Hybrid Linac-MRs

With the advent of clinical linac-MR systems, there has been considerable interest in the feasibility of markerless MR-based tumour tracking concurrent with irradiation. This presents several unique challenges: charged particles responsible for depositing dose can be deflected by the MRI's main magnetic field (B_0), potentially resulting in increased skin dose and dosimetric hot spots; RF noise from the linac and DMLCs can interfere with the MR acquisition process; MV radiation can induce current in metals producing additional RF noise; and MR images often exhibit spatial distortion (especially near regions

of with highly variable magnetic susceptibility, such as the lungs), making consistent tumour localization difficult.

Additionally, MR imaging has inherently longer acquisition times than most alternative tumour tracking methods (for example, fluoroscopy is typically performed around 30 Hz, while 2D real-time MR is usually in the 4-8 Hz range). This has two major implications: (1) a coarser sampling of the patient's respiratory cycles, which may obfuscate some of the more subtle features of the tumour motion; and (2) a longer system delay. Additionally, MR images can take much longer to reconstruct after acquisition, even further extending the system delay.

The Alberta Linac-MR[26] is currently being commissioned for MR-IGRT at the Cross Cancer Institute. It has a rotating biplanar configuration, which allows for a reduction in charged particle deflection since the beam can be consistently oriented along B_0. It additionally results in a far less claustrophobic environment for the patient than a traditional cylindrical bore MR, and allows for a greater degree of couch motion which enables isocentric treatments and avoids the need for adaptive planning for every patient. It also has relatively low B_0 strength (0.5T) compared to other linac-MRs that produces less geometric distortion, better CNR for some tumour types[49], further minimizes charged particle deflection and can be generated with a high-temperature superconducting magnet, allowing for the device to be installed in a traditional linac vault.

1.10 Non-invasive Intrafractional Tumour-Tracked Radiotherapy

Non-invasive intrafractional tumour-tracked radiotherapy (nifteRT)[50] is a novel DTTRT approach that is being developed on the Alberta Linac-MR. The proposed nifteRT workflow is illustrated in Figure 1.3.

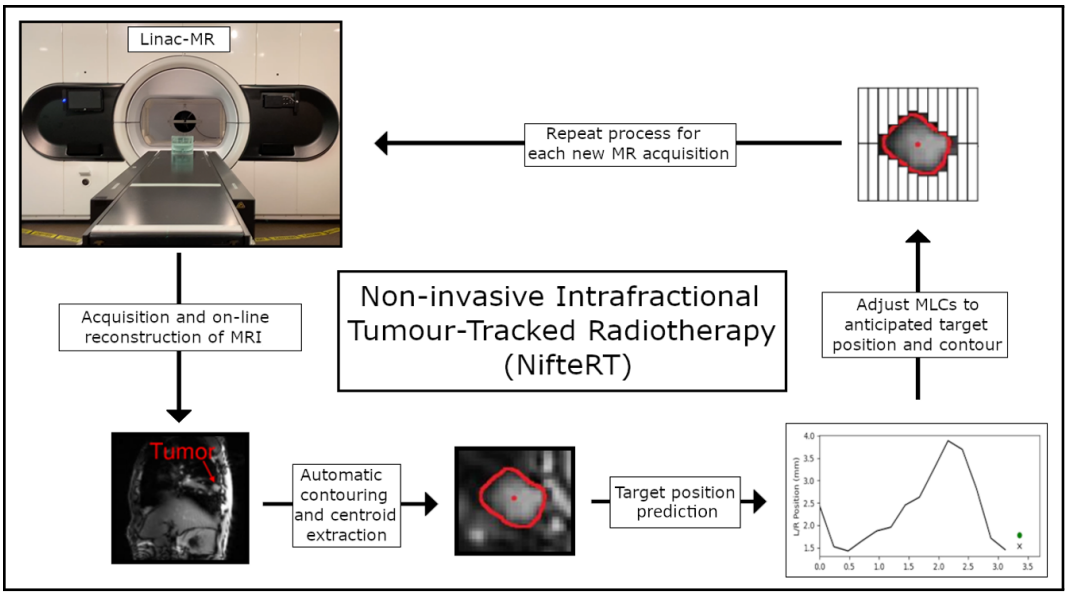

Figure 1.3: An illustration of the nifteRT workflow. Reproduced in adapted form with permission from J. Yun.

Briefly, MR imaging of the tumour is acquired at a frame rate of 4 Hz or greater. After reconstruction, the image is automatically contoured using a trained ANN and the centroid position of the tumour is calculated. A future centroid position is then predicted based on the tumour's recent motion history, and the current contour and anticipated future centroid are then sent to the DMLCs which immediately begin driving toward the target position.

The system delay for this entire process has been estimated at between 275 ms and 340 ms for typical lung tumour treatments [51]. This delay consists of (1) the effective acquisition time (one-half the total acquisition time for MR pulse sequences that cross the origin of k-space halfway through acquisition, see Section 2.2.2 for an explanation), (2) processing and reconstruction of the image, (3) autocontouring of the tumour and centroid determination, (4) motion prediction, (5) DMLC motion and (6) communication between devices. The effective image acquisition time is the dominant factor in determining system delay (for 4 Hz imaging, one-half of 250 ms yields 125 ms), followed by reconstruction and DMLC motion (both typically on the order of several tens

of ms, though the contribution of the DMLC motion depends on the maximum velocity of the tumour). Neural network-based autocontouring takes about 20 ms when optimized[52], as does the motion prediction algorithm outlined in Chapter 3.

1.11 Tumour Motion Prediction

The idea of predicting respiration-induced tumour motion has been around since at least 2002, initially intended to improve gated EBRT based on real-time imaging of fiducial markers [53]. It has since evolved considerably as increased computing power has made more sophisticated prediction methods viable. Initially, static linear and sinusoidal models, simple neural networks and Kalman filters were explored[53]–[55], resulting in modest improvements to tracking. Even at this early juncture, it was recognized that the selected approach to tumour motion prediction should depend on the system delay. That is, several methods (such as the normalized least mean squares approach used by the CyberKnife and Vero systems) perform well when the system delay is kept below 200 ms and the motion is well-approximated as linear, but their capabilities drastically fall off thereafter. Most of the subsequent literature therefore focused on the more difficult problem of non-linear motion prediction at long system delays.

Adaptive approaches, in which the predictor's internal parameters are re-calculated as new motion data become available, were shown to result in considerably better performance than their static counterparts [56], [57]. Model-free autoregressive approaches soon followed [58], [59], which performed better when patients exhibited irregular respiratory patterns [60].

Around 2008, there was a rapid development in the application of deep neural networks as a result of major machine learning competitions [61]. Over the next few years, these developments filtered into tumour motion prediction

research [62]–[66] resulting in more accurate predictions, but also increased computational expense.

Around the same time, more complex Kalman models [67], [68] were developed that could better address irregular breathing patterns, and support [69], [70] and relevance vector machines [71] provided performance comparable to the neural networks of the time but with considerably less computational expense. RNNs were later shown to outperform traditional feed-forward neural networks [50], [72], [73], since they are capable of responding to temporal patterns in sequential data (see Section 2.1.6 for more details).

Currently, there is no consensus on the optimal approach to tumour motion prediction for DTTRT, and it is difficult to draw comparisons between the different approaches presented in the literature. There are several reasons for this: (1) there is no standardized tumour motion dataset that is common to all publications; (2) researchers tend to focus on one clinical application of their approach, so they typically test a narrow range of acquisition times and system delays; (3) there is no standardized way to report predictive accuracy, and (4) each prediction method typically has an expansive set of free parameters, so a wide range of results are possible even under the "same" approach.

When studies comparing predictive accuracy across multiple categories of predictor are performed, they generally conclude that any method that is used for long system delays should at least be non-linear and adaptive. Further, with an eye toward an eventual clinical use, it should also be relatively quick and computationally inexpensive to implement. In this thesis, I develop an approach to neural network-based motion prediction that satisfies all of these criteria.

1.12 Research Motivation

Previously, the results of a study applying long short-term memory (LSTM, see Section 2.1.6) RNNs for 3D tumour motion prediction at an acquisition time of 280 ms and a prediction horizon of 280 ms were published[50]. A generic RNN structure (two hidden layers with 256 neurons each, see Section 2.1.2 for an explanation of these terms) was used for all patients, and a considerable amount of time (up to several hours) was required to train the networks. Based on these results, I was initially posed three questions to answer:

- Is there any benefit to customizing the architecture of the networks on a patient- or fraction-specific basis?
- What is the effect of varying acquisition rates and system delays on predictive accuracy?
- Can the training process be accelerated by using graphics processing units (GPUs), which are intended for handling the large tensor calculations required for neural network training?

Over time, these questions evolved as I experimented with new network architectures and training processes, and I eventually landed on a considerably different approach to motion prediction than anything that had been previously published. However, the underlying goals of improving the accuracy, reliability and speed of neural network-based tumour motion prediction remained, and the work presented in Chapter 3 represents a realization of all three.

Chapter 2

Theory

2.1 Artificial Neural Networks

In Chapter 1, the need for a non-linear method for tumour motion prediction when using hardware with long system delays ($>$ 200 ms, which includes the Alberta Linac-MR) was identified. ANNs are computational models that can be "trained" to perform arbitrary non-linear tasks. They are inspired by the neural networks of biological organisms both in terms of their architectural layout and the functioning of their component neurons, though there exist some important distinctions.

2.1.1 The Neuron

In biological systems, neurons are cells found in the central and peripheral nervous systems that are responsible for generating, carrying and processing electrical signals associated with movement, sensation and cognition. Their basic function is to sum the incoming electrical and chemical signals from other cells in the form of electrical potentials and, if a pre-determined threshold potential is reached, to generate and pass along an electrical impulse called an action potential (see Figures 2.1(a) and (b)). As a simple mathematical model, if V_a is the magnitude of the action potential, V_T is the threshold potential of a neuron, the output of the i^{th} upstream cell is given by V_i and the connection

strengths that determine how much of each incoming signal make it to the neuron is w_i, then the output of the neuron V_o is given by:

$$V_o = V_a \Theta \left(\sum_i w_i V_i - V_T \right) \tag{2.1}$$

where Θ is the Heaviside function:

$$\Theta(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases} \tag{2.2}$$

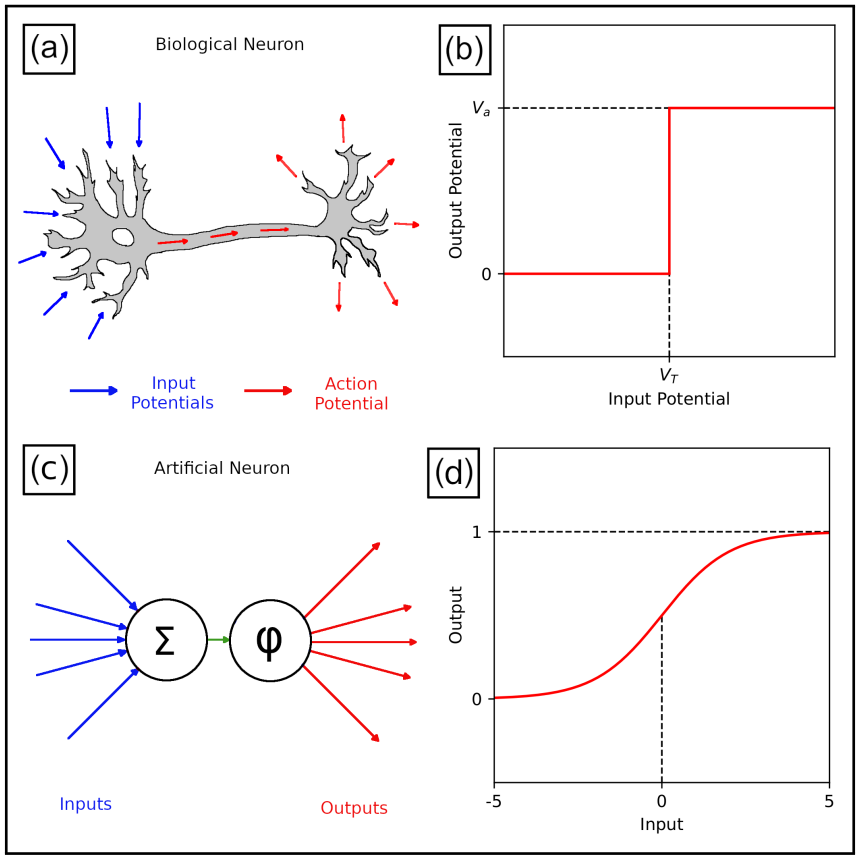

Figure 2.1: (a) and (b): A simplified schematic of a biological neuron and its input/output relationship, respectively. (c) and (d): A schematic of an artificial neuron and its input/output relationship (assuming a sigmoid activation function), respectively. Interestingly, the biological neuron is the digital one while the artificial neuron is analog.

Despite this relatively simple functioning at the level of the individual neuron, biological organisms are capable of extremely complex and adaptable

behaviour. This is because of the sheer number of neurons and connections in the brain (estimated to be 10^{10} and 10^{15} for humans, respectively[74]) as well as the ability to forge new connections (neural plasticity[75]) and tune their strength (synaptic plasticity[76], equivalent to adjusting w_i in Equation 2.1) as a response to repeated activation.

An artificial neuron is similar in that it receives inputs either externally or from other artificial neurons, performs some mathematical function on those inputs, and then generates an output (see Figure 2.1(c)). However, the mathematics of an artificial neuron can be made more sophisticated than the all-or-nothing activation of a biological neuron (Figure 2.1(d)). This allows for the representation of complex functions using fewer neurons, and also facilitates the training of ANNs through gradient descent (see Section 2.1.3).

If the inputs to an artificial neuron are denoted x_i, the strengths of the connections between the inputs and the artificial neuron are w_i, and the output of the neuron is y, then

$$y = \phi\left(\Sigma\left(w_i x_i\right)\right) \tag{2.3}$$

Here, Σ is called the artificial neuron's summation function, though it does not necessarily represent a simple summation of the weighted inputs $w_i x_i$. For example, in max pooling stages in convolutional neural networks, it instead selects only the maximum weighted input value. However, for simplicity and because a truly summative summation function will be used in all the networks presented in this thesis, henceforth it will be assumed that the summation function represents a direct summation of the weighted inputs.

ϕ is called the neuron's activation function, and it can be selected from a range of linear and non-linear functions depending on the neuron's specific purpose within the ANN. As is the case with biological neurons, an artificial neuron on its own is not capable of performing sophisticated tasks – this ability

emerges only when multiple neurons are connected together.

2.1.2 Networks of Neurons

Biological neural networks are extremely complex, with circuitous paths between neurons, two-way communication channels and neuronal self-connections. In order to make ANNs easier to define and implement, their artificial neurons are typically organized into discrete *layers*. Moreover, these layers are generally *fully connected* – that is, each neuron receives inputs from every neuron in the previous layer, and sends a copy of its output to every neuron in the subsequent layer. Finally, for simplicity, every neuron in a given layer generally has identical summation and activation functions.

Figure 2.2 depicts two fully-connected neuron layers, the first containing m neurons and the second containing n neurons. All the outputs of the neurons in the first layer can be represented as an m-vector $\vec{X}$, and all the weights connecting the i^{th} neuron in the first layer to the j^{th} neuron in the second layer (w_{ij}) can together be represented as an $m \times n$ matrix $\mathbf{W}$:

$$\vec{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}, \qquad \mathbf{W} = \begin{bmatrix} w_{11} & \dots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{m1} & \dots & w_{mn} \end{bmatrix} \tag{2.4}$$

An n-vector containing the weighted and summed inputs for each neuron in the second layer is then given by the weight matrix applied to the outputs of the first layer. It is convenient to call this vector $\vec{Z}$, with $\vec{Z} = \mathbf{W}^T \vec{X}$ where ^{T} is the matrix transposition operator, or

$$\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} = \begin{bmatrix} w_{11} & \dots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{m1} & \dots & w_{mn} \end{bmatrix}^T \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} \tag{2.5}$$

The activation function ϕ for the second layer then acts on this vector to

give the output of the second layer $\vec{Y}$, which is itself also an n-vector. All together, $\vec{Y} = \phi(\vec{Z}) = \phi(\mathbf{W}^T \vec{X})$, or

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \phi \left(\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} \right) = \phi \left(\begin{bmatrix} w_{11} & \dots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{m1} & \dots & w_{mn} \end{bmatrix}^T \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} \right) \tag{2.6}$$

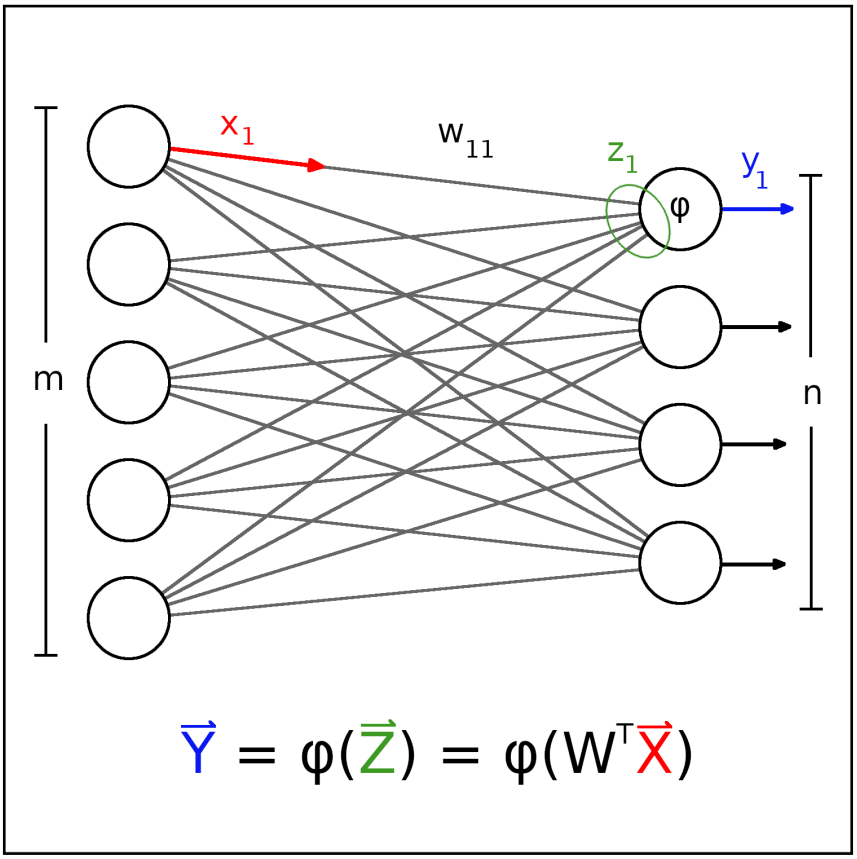

Figure 2.2: An example of two fully-connected layers in an ANN.

Generally, ANNs consist of three main components, as indicated in Figure 2.3: (1) an input layer, (2) any number of computing layers (also known as hidden layers), and (3) an output layer. Typically, the computing layers also contain what are called *bias neurons* attached to them. These are neurons that do not take in any input and always output +1, which is then fed through weighted connections to the subsequent layer. In effect, these bias nodes apply an adjustable shift to the zero point of the activation functions of the down-

stream neurons. This is important, since most activation functions have the property $\phi(0) = 0$, but the non-linear process intended to be emulated may not. Many authors choose to explicitly differentiate bias nodes from the rest of the computing neurons in their notation, but in reality they can simply be treated like any other artificial neuron mathematically.

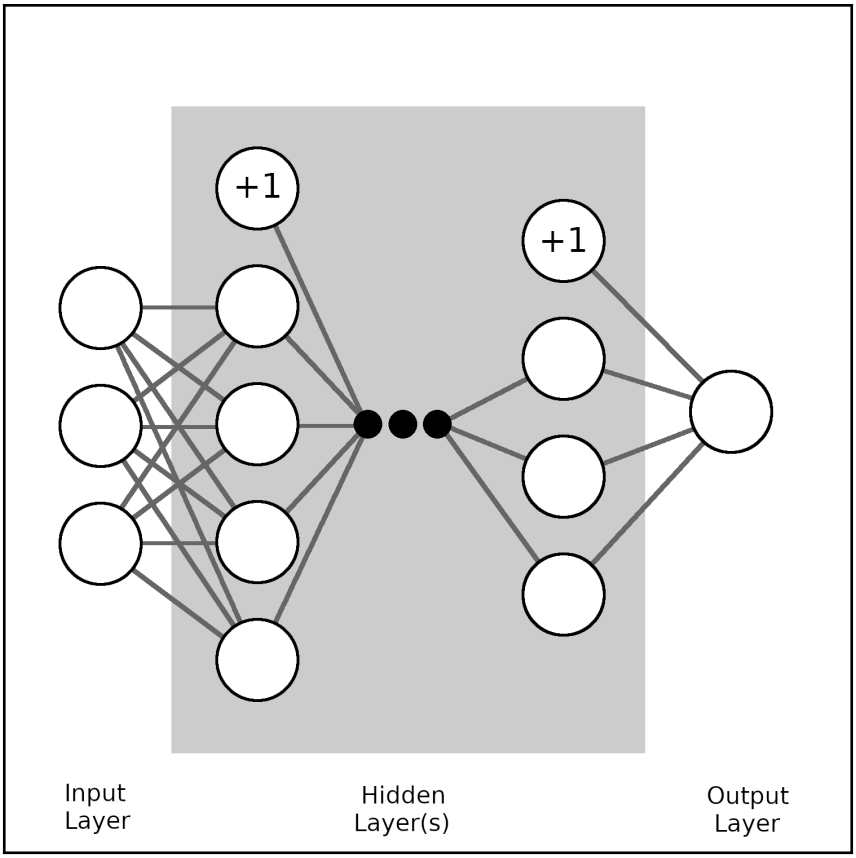

Figure 2.3: The general structure of an ANN, with an input layer, any number of hidden layers, and an output layer. In this illustration, the input layer has a width of 3, the first hidden layer a width of 4 with a bias neuron, the last hidden layer a width of 3 with a bias neuron, and the output layer a width of 1. The ellipsis represents the potential for many more hidden layers that are not shown.

A very simple ANN is depicted in Figure 2.4. It has an input layer consisting of two neurons, a hidden layer consisting of two neurons with no bias, and an output layer consisting of one neuron. In order to better differentiate between the layers, the inputs to the network will be denoted $\vec{Y}^1$, the four

weights connecting the input neurons to the hidden layer neurons comprise the weight matrix $\mathbf{W}^1$, the weighted inputs to the hidden layer neurons are $\vec{Z}^1$, the activation function of the hidden layer is ϕ^2, the two outputs of the hidden layer are contained in $\vec{Y}^2$, the weight matrix connecting the hidden layer to the output layer is $\mathbf{W}^2$, the weighted inputs to the output layer are $\vec{Z}^2$, the activation function of the output layer is ϕ^3, and the final output of the network is a single value, Y^3. Mathematically,

$$Y^3 = \phi^3(\vec{Z}^2) = \phi^3((\mathbf{W}^2)^T\vec{Y}^2) = \phi^3((\mathbf{W}^2)^T\phi^2(\vec{Z}^1)) = \phi^3((\mathbf{W}^2)^T\phi^2((\mathbf{W}^1)^T\vec{Y}^1)) \tag{2.7}$$

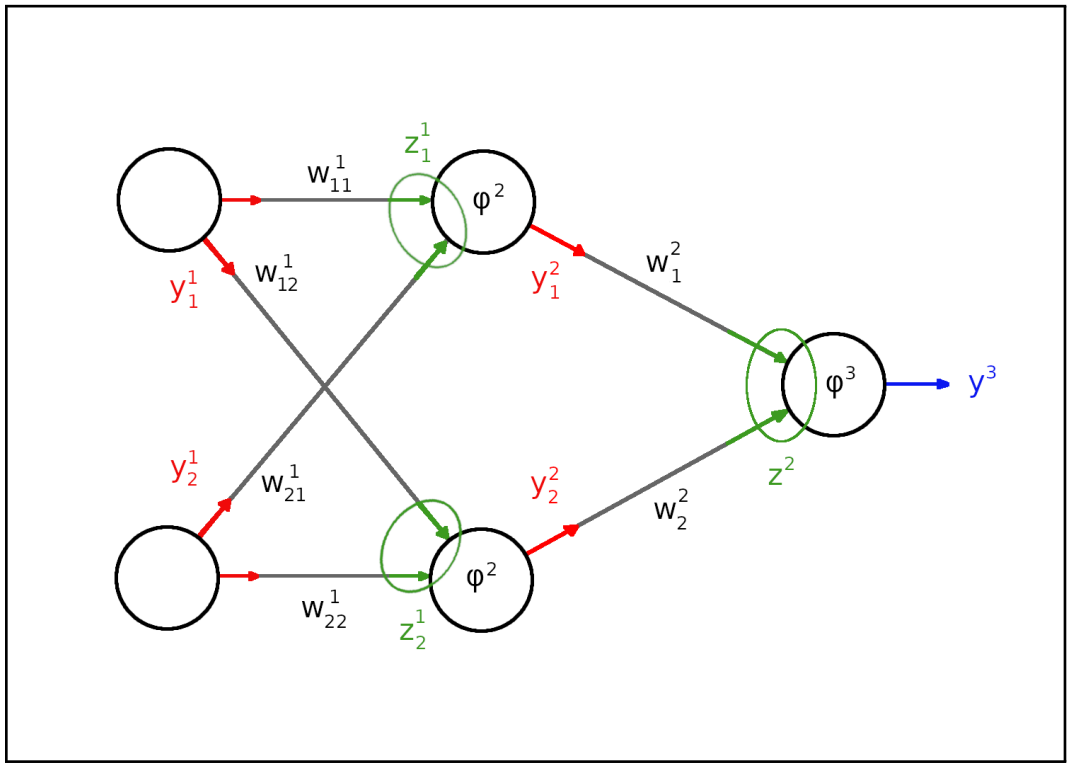

Figure 2.4: A simple ANN used to demonstrate their basic function and training.

Generally, the output of an N-layered ANN is given by:

$$Y^N = \phi^N\Big((\mathbf{W}^{N-1})^T\phi^{N-1}\Big((\mathbf{W}^{N-2})^T \ldots \phi^2((\mathbf{W}^1)^T\vec{Y}^1)\Big)\Big) \tag{2.8}$$

There are two brief asides that should be mentioned here: First, if all of the activation functions in the simple ANN are linear (i.e. $\phi^i(z) = k^i z$ with k^i being a constant coefficient), then it can be shown that

$$Y^3 = k^3 k^2 \left((\mathbf{W}^2)^T \left((\mathbf{W}^1)^T \vec{Y}^1 \right) \right) = \left(k^3 k^2 (\mathbf{W}^2)^T (\mathbf{W}^1)^T \right) \vec{Y}^1 \tag{2.9}$$

If the multiplication of the matrices is evaluated first, and the weight that connects the i^{th} neuron in layer k to the j^{th} neuron in layer $k+1$ is denoted w_{ij}^k:

$$\begin{aligned} (\mathbf{W}^2)^T (\mathbf{W}^1)^T &= \begin{bmatrix} w_{11}^2 & w_{21}^2 \end{bmatrix} \begin{bmatrix} w_{11}^1 & w_{21}^1 \\ w_{12}^1 & w_{22}^1 \end{bmatrix} \\ &= \begin{bmatrix} w_{11}^2 w_{11}^1 + w_{21}^2 w_{12}^1 & w_{11}^2 w_{21}^1 + w_{21}^2 w_{22}^1 \end{bmatrix} \end{aligned} \tag{2.10}$$

However, $w_{11}^2 w_{11}^1 + w_{21}^2 w_{12}^1$ and $w_{11}^2 w_{21}^1 + w_{21}^2 w_{22}^1$ are just complicated ways of representing any arbitrary pair of numbers A and B. If the l^{th} input is y_l^1, then the equation describing the entire network is now:

$$Y^3 = \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} y_1^1 \\ y_2^1 \end{bmatrix} = A y_1^1 + B y_2^1 \tag{2.11}$$

That is, this ANN is just an unnecessarily complicated linear function. This argument can be extended to an N-layered network with all linear activation functions and any number of outputs and inputs:

$$Y^N = \left(\prod_{i=2}^{N} k^i \right) \left(\prod_{i=1}^{N-1} (\mathbf{W}^i)^T \right) \vec{Y}^1 \tag{2.12}$$

If at least one ϕ^i is non-linear, however, the network represents a non-linear transformation between inputs and outputs, which is what is required for motion prediction at long system delays.

Second, if the input data are multi-dimensional (for example, the 3D spatial tumour coordinates considered in the manuscript in Chapter 3), extra

dimensions can be added to the weight matrices so that all of the input dimensions independently contribute to the output. This is equivalent to having multiple parallel networks all joined at the output nodes (see Figure 2.5). Multi-dimensional outputs are also possible. However, in the case of the motion prediction application outlined in Chapter 3, the output is a single 3D coordinate, which can be represented as an output layer with three neurons. In this way, previous motion from all three input dimensions (S/I, A/P, L/R) are used to predict the motion in each output dimension.

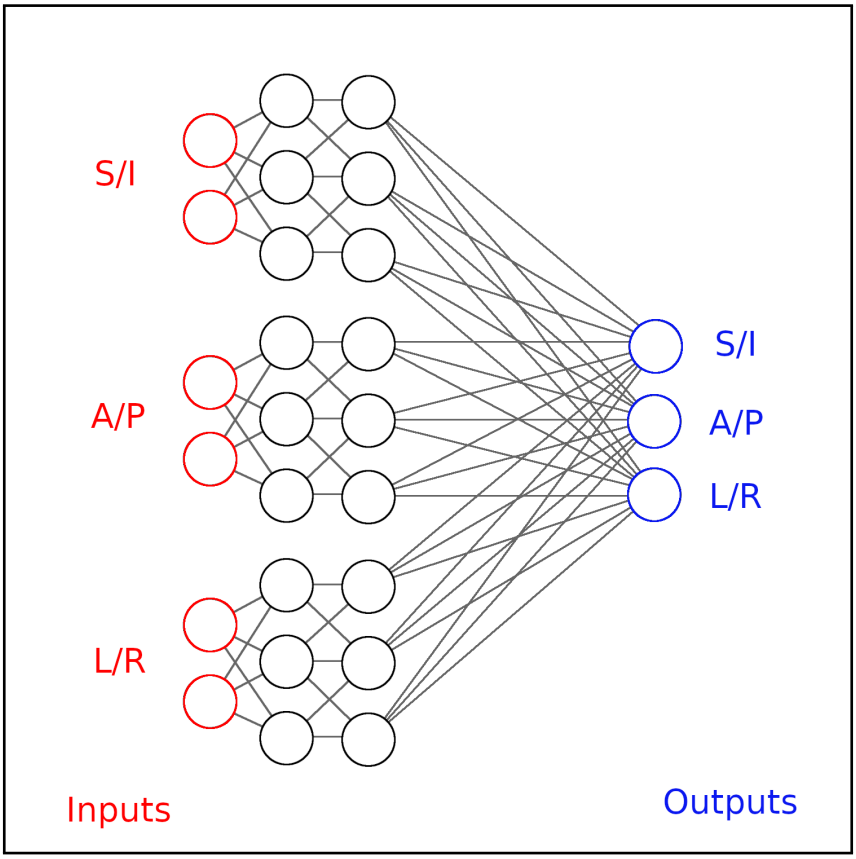

Figure 2.5: A parallel network diagram showing how three-dimensional inputs, in this case motion data in the three cardinal patient directions, can each be used to predict the future motion in any given direction. For illustrative purposes, simple networks with two input nodes and two hidden layers each of width three are shown, while much more complex network structures will be used in Chapter 3.

In this section, a method has been simply outlined for constructing an ANN

(and, therefore, a non-linear transformation) of arbitrary complexity. How well an ANN *possibly can* approximate a given non-linear process is governed by the number of hidden layers (also called the *depth* of the network), the number of neurons in each hidden layer (the layer's *width*), and the chosen activation function for each layer. That is, these parameters define its *capacity*. How well it actually does approximate that process is determined by the values in the weight matrices connecting the layers. That is, these weight matrices are where the task-specific "knowledge" is stored.

This has parallels to biological neural networks – a human can be born with the capacity to eventually read, walk and talk, but it takes time, practice and repeated exposure to fine-tune the neuronal connections that will allow them to do those things well. In the next section we will introduce how ANNs are taught to emulate non-linear processes through iterative optimization of their internal weights.

2.1.3 Supervised Learning, Backpropagation and Gradient Descent

Supervised learning is the conventional method for teaching an ANN to perform a desired task, and it requires two things: (1) examples of that task being done correctly (called the *training set*), consisting of paired inputs and their corresponding ground-truth outputs; and (2) a metric that compares the output of the network to the ground-truth to quantify the network's performance. This metric is called the cost function, C. The basic goal of supervised learning is to find a set of network weights that minimize C by iteratively repeating two steps, forward propagation and backpropagation.

Forward propagation simply entails feeding an input from the training set into the network, and calculating the output using Equation 2.8. Importantly, during the forward propagation process the results of intermediate calculations

should be stored for future reference. Once a network output Y^N is calculated, its cost function C can be computed using the true value $\hat{Y}$ from the training data.

During backpropagation, the gradient of C with respect to each weight in the ANN is computed. Using the simple network illustrated in Figure 2.4 and mathematically described in Equation 2.7 as an example, the cost function gradient with respect to the total weight matrix $\mathbf{W}^2$ can be calculated via the chain rule, as follows:

$$\frac{\partial C}{\partial \mathbf{W}^2} = \frac{\partial C}{\partial Y^3}\frac{\partial Y^3}{\partial \vec{Z}^2}\frac{\partial \vec{Z}^2}{\partial \mathbf{W}^2} \tag{2.13}$$

Since $Y^3 = \phi^3(\vec{Z}^2)$, then

$$\frac{\partial Y^3}{\partial \vec{Z}^2} = \frac{\partial}{\partial \vec{Z}^2}\phi^3\left(\vec{Z}^2\right) \tag{2.14}$$

where

$$\vec{Z}^2 = (\mathbf{W}^2)^T\vec{Y}^2 \tag{2.15}$$

Now, since $\mathbf{W}^2$ is actually just a vector in this instance, the definition of the gradient vector is required:

$$\frac{\partial f(\vec{x})}{\partial \vec{x}} = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_N} \end{bmatrix} \tag{2.16}$$

Since

$$\vec{Z}^2 = \begin{bmatrix} w_1^2 & w_2^2 \end{bmatrix}\begin{bmatrix} y_1^2 \\ y_2^2 \end{bmatrix} = w_1^2 y_1^2 + w_2^2 y_2^2 \tag{2.17}$$

then

$$\frac{\partial \vec{Z}^2}{\partial \mathbf{W}^2} = \begin{bmatrix} \frac{\partial \vec{Z}^2}{\partial w_1^2} \\ \frac{\partial \vec{Z}^2}{\partial w_2^2} \end{bmatrix} = \begin{bmatrix} y_1^2 \\ y_2^2 \end{bmatrix} \tag{2.18}$$

This leaves a useful expression for the partial derivative of the cost function with respect to the matrix of weights connecting the hidden layer to the output layer:

$$\frac{\partial C}{\partial \mathbf{W}^2} = \frac{\partial C}{\partial Y^3} \left(\frac{\partial}{\partial \vec{Z}^2} \phi^3 \left(\vec{Z}^2 \right) \right) \vec{Y}^2 \tag{2.19}$$

Most cost functions are selected such that they have the property

$$\frac{\partial C}{\partial Y^N} = f(C) \tag{2.20}$$

so the first term is trivial to compute from value of C that was found and stored during forward propagation.

A broad variety of activation functions are available for both the hidden and output layers, with some common choices illustrated in Figure 2.6. For the output layer, the activation function should be selected such that its limits match those of the desired output. For example, a neural network that classifies images might output a "confidence level" in the range of 0% – 100% that a certain object appears in the image. In this case, the output should be constrained since a negative or $> 100\%$ confidence would be meaningless, so an output activation function resembling the sigmoid or tanh functions is often used (with appropriate scaling). For the example of tumour motion prediction, there are no well-defined constraints on the tumour's position, so a linear output function would be appropriate. For hidden layer activation functions, the rationale behind choosing a particular activation function is less clear, and often comes down to trial and error.

Regardless, activations functions are typically selected such that they either have trivial gradients (as is the case with ReLU, where the gradient itself is

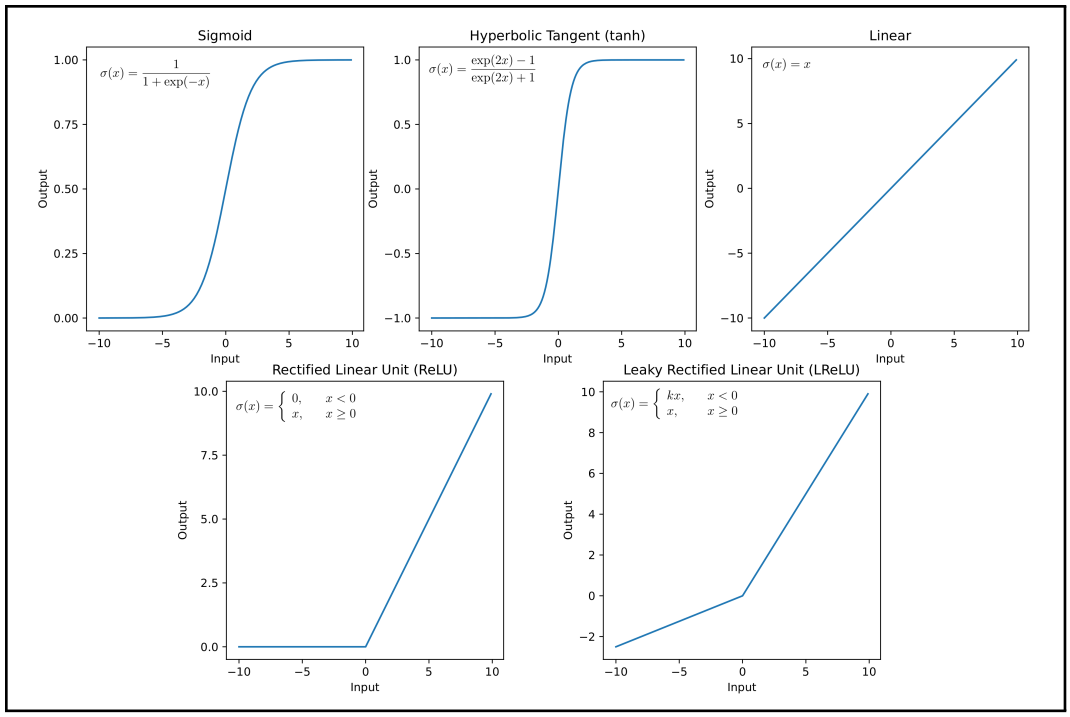

Figure 2.6: An illustration of common activation functions used for hidden and output layers in ANNs, including the sigmoid, hyperbolic tangent (tanh), linear, rectified linear unit (ReLU) and leaky ReLU (LReLU) functions.

just the Heaviside function), or have the property:

$$\frac{\partial}{\partial z}\phi(z) = f(\phi(z)) \tag{2.21}$$

such as the tanh activation function, for which

$$\frac{\partial}{\partial z}\tanh(z) = 1 - \tanh^2(z) \tag{2.22}$$

In the latter case, $\phi^3(\vec{Z}^2)$ is just Y^3, and along with $\vec{Y}^2$ it was computed and stored during the forward propagation step, so it does not need to be re-calculated. Finding the partial derivative of C with respect to $\mathbf{W}^2$ is therefore computationally inexpensive, given appropriate forms for C and ϕ.

This approach can be extended for $\mathbf{W}^1$, the weight matrix connecting the input to the hidden layer:

$$\frac{\partial C}{\partial \mathbf{W}^1} = \frac{\partial C}{\partial Y^3}\frac{\partial Y^3}{\partial \vec{Z}^2}\frac{\partial \vec{Z}^2}{\partial \vec{Y}^2}\frac{\partial \vec{Y}^2}{\partial \vec{Z}^1}\frac{\partial \vec{Z}^1}{\partial \mathbf{W}^1} \tag{2.23}$$

The first two terms have already been determined while calculating the cost function gradient with respect to $\mathbf{W}^2$. From Equation 2.15,

$$\frac{\partial \vec{Z}^2}{\partial \vec{Y}^2} = (\mathbf{W}^2)^T \tag{2.24}$$

Similar to Equation 2.14,

$$\frac{\partial \vec{Y}^2}{\partial \vec{Z}^1} = \frac{\partial}{\partial \vec{Z}^1}\phi^2(\vec{Z}^1) \tag{2.25}$$

which again can be quickly solved using the values calculated in the forward propagation step. Finally, since $\mathbf{W}^1$ is a matrix this time, in order to calculate $\partial \vec{Z}^1/\partial \mathbf{W}^1$ the definition of the Jacobian matrix is required. If $\vec{\nu}$ is a column m-vector such that:

$$\vec{\nu} = \begin{bmatrix} \nu_1 \\ \nu_2 \\ \vdots \\ \nu_m \end{bmatrix} \tag{2.26}$$

then

$$\frac{\partial \vec{\nu}(\mathbf{X})}{\partial \mathbf{X}} = \begin{bmatrix} \frac{\partial \nu_1}{\partial x_{11}} & \cdots & \frac{\partial \nu_1}{\partial x_{1n}} \\ \vdots & \ddots & \vdots \\ \frac{\partial \nu_m}{\partial x_{m1}} & \cdots & \frac{\partial \nu_m}{\partial x_{mn}} \end{bmatrix} \tag{2.27}$$

Since

$$\vec{Z}^1 = \begin{bmatrix} w_{11}^1 & w_{21}^1 \\ w_{12}^1 & w_{22}^1 \end{bmatrix}\begin{bmatrix} y_1^1 \\ y_2^1 \end{bmatrix} = \begin{bmatrix} w_{11}^1 y_1^1 + w_{21}^1 y_2^1 \\ w_{12}^1 y_1^1 + w_{22}^1 y_2^1 \end{bmatrix} \tag{2.28}$$

then

$$\frac{\partial \vec{Z}^1}{\partial \mathbf{W}^1} = \begin{bmatrix} y_1^1 & 0 \\ 0 & y_2^1 \end{bmatrix} \tag{2.29}$$

More generally, for an N-layered network, the gradient of the cost function with respect to the i^{th} weight matrix $\mathbf{W}^i$ is

$$\frac{\partial C}{\partial \mathbf{W}^i} = \frac{\partial C}{\partial Y^N}\frac{\partial Y^N}{\partial \vec{Z}^{(N-1)}}\frac{\partial \vec{Z}^{(N-1)}}{\partial \vec{Y}^{(N-1)}}\cdots\frac{\partial \vec{Y}^{(i+1)}}{\partial \vec{Z}^i}\frac{\partial \vec{Z}^i}{\partial \mathbf{W}^i} \tag{2.30}$$

Each of these terms can be obtained relatively quickly by plugging intermediate results from the forward propagation step into the equations:

$$\frac{\partial C}{\partial Y^N} = f(C) \tag{2.31}$$

$$\frac{\partial \vec{Y}^k}{\partial \vec{Z}^{(k-1)}} = \frac{\partial}{\partial \vec{Z}^{(k-1)}}\phi^k(\vec{Z}^{(k-1)}) = f\left(\phi^k(\vec{Z}^{(k-1)})\right) \tag{2.32}$$

$$\frac{\partial \vec{Z}^k}{\partial \vec{Y}^k} = (W^k)^T \tag{2.33}$$

$$\frac{\partial \vec{Z}^i}{\partial \mathbf{W}^i} = \vec{Y}^i\mathbf{I}_m \tag{2.34}$$

where $\mathbf{I}_m$ is an $m \times m$ identity matrix.

However, this gradient only stems from a single input/output pair. During training, the mean cost function evaluated over the entirety of the training set, $\bar{C}$, should be considered when updating weights. If there are μ input/output pairs in the training set, then:

$$\bar{C} = \frac{1}{\mu}\sum_{i=1}^{\mu} C_i \tag{2.35}$$

If each individual weight is then updated according to the equation:

$$w_{i,j}^k := w_{i,j}^k - \alpha\frac{\partial \bar{C}}{\partial w_{i,j}^k} \tag{2.36}$$

then the entire ANN should simultaneously step in a direction that will reduce $\bar{C}$ during the next forward propagation (even though it might not be in the

right direction to reduce C for each individual input/output pair in the training set).

Here, α is called the learning rate of the training process, and it acts as a scaling factor for the step size in weight space. It can either be a fixed value, scheduled in advance or dynamically updated based on the measured cost function by an optimizer algorithm. However, as will be shown in Chapter 3, it is often much more consequential than just determining the rate of training.

A single cycle of forward propagation, storage of intermediate results, back-propagation of error using those intermediate results and corrective weight adjustments based on the full training set is collectively called a *training epoch*. At the beginning of training, the weights in the network are usually randomly initialized. Training epochs are then repeated until $\bar{C}$ approaches a stable minimum value. For particularly large training sets, it might be advantageous in terms of training time to update the internal weights more often than once per epoch. This is called *mini-batch gradient descent*, and an extreme example of this approach, where a weight update is performed after every single training example, is called *stochastic gradient descent*. In the application considered in this thesis, fraction-specific motion prediction, there are typically only $\sim 10^2$ motion examples in the training set, so batch training of this form is not required.

A chosen cost function should have several properties to help facilitate the ANN training process. First, it should ideally be convex, so that the gradient always points toward a single global minimum. It should also be positive definite, so that the cost functions of two different training examples cannot cancel each other out. Finally, it should be differentiable, so that $\partial\bar{C}/\partial Y^N$ is always defined. It is not absolutely necessary for $\partial C/\partial Y^N$ to be representable as some function of C, but it speeds up the backpropagation step if this is the case.

Figure 2.7(a) shows a simplified illustration how the training process typically proceeds. A cost function $\bar{C}$ taken over the entire training set is plotted as a function of a single network weight. In case A, the learning rate α is quite small, and the training process takes a large number of epochs to reach the vicinity of the minimum $\bar{C}$. As α is increased in case B, the rate at which the network approaches the minimum $\bar{C}$ is increased. However, if it is too large then the weights will oscillate about their optimal values during convergence and slow down training, as seen in case C. For a larger α still in case D, $\bar{C}$ can grow between epochs resulting in weight divergence. Figure 2.7(b) shows an alternative illustration of this concept, plotting $\bar{C}$ as a function of training epoch for a range of α values corresponding to cases A through D.

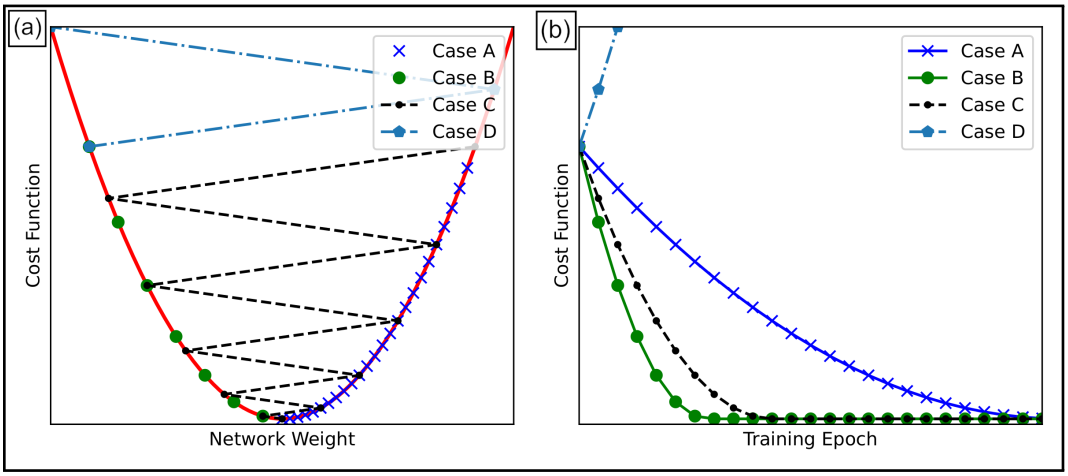

Figure 2.7: (a): A simplified illustration of how the cost function may vary with a single network weight, and how training may proceed (or fail) depending on the learning rate used for gradient descent. Cases A through D illustrate increasingly larger learning rates, starting from an inefficiently small learning rate (A), an appropriate one (B), one that results in oscillation about the global minimum (C) and one that leads to divergence (D). (b) Another representation of (a), showing cost function as a function of training epoch for the various learning rates.

In reality, training is much more complicated than this simple illustration can express. ANNs can have thousands, millions or even hundreds of billions[77] of trainable parameters that are often interdependent. Therefore,

the gradient descent process is not guaranteed to reduce $\bar{C}$ after each epoch, and the $\bar{C}$ that the ANN eventually converges to is not necessarily its global minimum value. That is, it can get trapped in one of any number of local minima that arise due to interactions between network weights. As a result, the solution that the training process yields (and therefore the performance of the ANN) in part depends on the random values assigned to the weights at the outset of training. The degree to which this instability manifests depends on a number of factors that will be discussed in Section 2.1.5.

It is also important to keep in mind that this process of supervised learning only results in an optimal set of ANN weights for the *training* data set, and that these weights are not guaranteed to be optimal when the ANN is applied to new data. There are many factors that affect how well trained ANNs generalize, which will also be discussed in more detail in Section 2.1.5.

2.1.4 Neural Network Hyperparameters and Hyperparameter Optimization

With an algorithm in place for training ANNs, the next step is to choose their structure and function in a way that yields optimal performance. ANNs are defined by their hyperparameters, which can be roughly divided into three categories: (1) those that determine the structural characteristics of the neural network (e.g., the number of hidden computing layers and the width of each layer) , (2) those that determine the functional or mathematical characteristics of the neural network (e.g., the activation functions at each layer), and (3) those that govern the supervised learning process (e.g., learning rate, optimizer, and the number of training epochs).

These ANN hyperparameters can be represented as a tuple of variables, some continuous (such as learning rate), some discrete (such as the number of hidden layers) and some categorical (such as the chosen activation function).

Hyperparameter optimization (HPO) is the process of finding the optimal values for each variable, in terms of minimizing a mean cost function $\bar{C}$.

There are several methods that can be employed when performing HPO. The simplest is a grid search algorithm, in which a set of reasonable bounds are determined for each hyperparameter and $\bar{C}$ is exhaustively calculated for every possible combination therein. Plotting $\bar{C}$ as a function of each hyperparameter averaged over the rest of hyperparameter space allows for the importance of each hyperparameter to be assigned. An important hyperparameter will have an observable relationship to $\bar{C}$ over its range, while an unimportant one will have a negligible effect on $\bar{C}$ (see Figure 2.8(a)). The optimal set of hyperparameters can be assigned as the combination that resulted in the smallest observed $\bar{C}$, or it may be interpolated from the plots of important hyperparameters versus $\bar{C}$ (and then usually tested for confirmation).

How the reasonable bounds for each hyperparameter are assigned and how densely to sample within these reasonable bounds is an important detail. It is entirely possible to leave the optimal hyperparameters outside of the range of the search, or to sample so coarsely that the minimum of the loss topography in hyperparameter space is missed. This must be balanced against the fact that casting a wider net with a finer mesh results in drastically increased computational expense, given the dimensionality of the HPO problem.

Another relatively straightforward HPO option is the random search algorithm[78], which is similar to the grid search except that hyperparameters are allowed to vary randomly rather than aligning to predefined grid points (see Figure 2.8(b)). This effectively allows for a finer sampling of hyperparameter space for the most important hyperparameters, but with the drawbacks of (1) more difficult disentangling of the effects of multiple important hyperparameters, and (2) the potential for a large portion of hyperparameter space to go randomly unsearched.

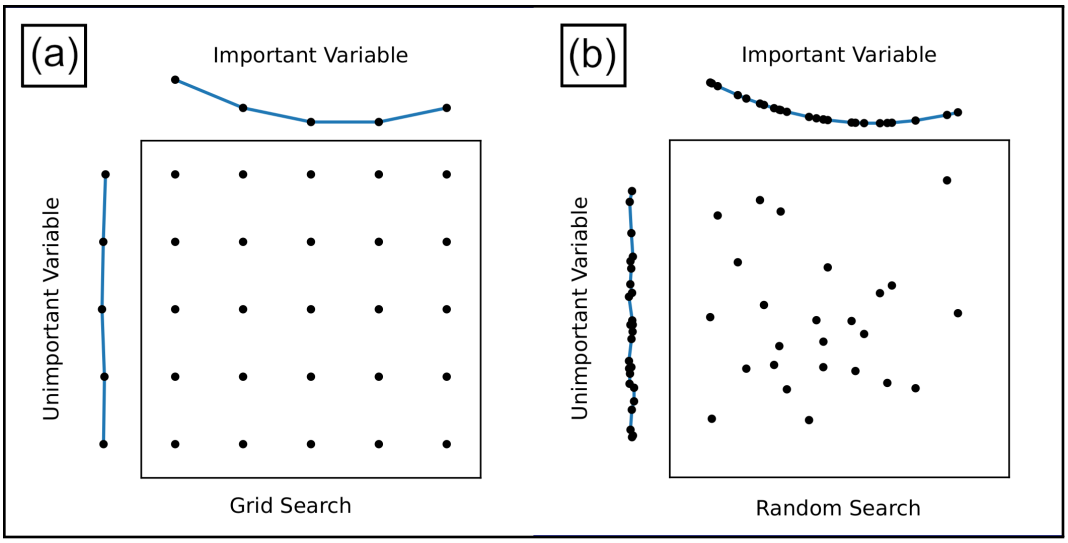

Figure 2.8: (a) Grid-search HPO. Hyperparameters (here, only two for illustration purposes) are varied regularly between reasonable bounds. The cost function (blue) will vary smoothly and appreciably with an important hyperparameter, while it will not appear to vary or will vary randomly with an unimportant one. (b) Random search HPO, which results in a more dense sampling of the cost function in hyperparameter space. Because this is an idealized illustration, the increased noise in the cost function plot that will be the result of less averaging is not shown.

Some of the more sophisticated HPO techniques are inspired by natural processes such as swarm dynamics and evolution. In particle swarm optimization[79], multiple "particles" are randomly initialized both in terms of their position and velocity in hyperparameter space, and their subsequent trajectories are determined by their own observations of the local $\bar{C}$ terrain as well as those of the swarm. In the covariance matrix adaptation evolution strategy[80], "organisms" with a random initial distribution of hyperparameters are allowed to mathematically reproduce based on their "evolutionary fitness" (their calculated $\bar{C}$), with random variations introduced to allow for the population to drift toward a more optimal solution. These strategies have an advantage in that they do not require a predetermined range to be assigned to each hyperparameter (although practically, the initialization of the particles or organisms in hyperparameter space should still be governed by prior

knowledge), but neither has a straightforward method for handling categorical variables.

Generally, HPO is a computationally demanding and lengthy process. Depending on the range of hyperparameter space being searched, it could easily require the evaluation of hundreds of thousands of different hyperparameter configurations. Worse yet, since initial weights cannot be made consistent across differently structured ANNs, in applications with a strong initial weight dependence multiple networks must be trained at each hyperparameter grid point to disentangle the effects of changing hyperparameters from the randomizing effect of the initial weights. A judgement must therefore be made balancing the anticipated benefits of HPO in terms of accuracy versus the time and effort it requires to perform.

For the specific case of respiration-induced tumour motion prediction, there are three choices for HPO:

- Determine a universal set of optimal hyperparameters for the task at hand, then use those hyperparameters for every patient and fraction thereafter
- Use a patient's past motion data to determine a patient-specific set of optimal hyperparameters
- Perform HPO prior to each treatment fraction using motion data measured that day

The first option, universal HPO, has the distinct advantage of only requiring one computationally expensive HPO to ever be performed. However, it is also likely to be the worst in terms of predictor accuracy, since it is unlikely that the same hyperparameter configuration will be optimal for every patient and fraction. The second, patient-specific HPO, only requires one optimization to be performed *per patient*. However, since it is unlikely that a full HPO

could be performed while the patient remains on the treatment couch, it would probably require each patient to be present for a purely observational mock treatment fraction. The final option, fraction-specific HPO, would fully account for how the day-to-day variability of the patient's respiratory patterns might affect the optimal ANN hyperparameters, but is the most computationally expensive and the most clinically impractical.

2.1.5 Overfitting and Other Challenges Inherent to "Small Data"

It might at first glance be reasonable to wonder why HPO is required at all. It is obvious that too simple of a network will struggle to carry out complex tasks regardless of the quality of its training, much like asking a fruit fly to prepare one's taxes will probably end in an audit no matter how much experience it claims to have. But can a network be too capacious? If not, the ANN could be made as large as the memory of the computing hardware allows, and the number of training epochs could be maximized based on the time available for training. However, it will be shown in this section that while this might be a valid strategy for obtaining the best possible *training* accuracy, it is in no way guaranteed to result in acceptable real-world performance.

The main reason for this is the finite nature of the training set – it contains only a limited number of examples of the kind of patterns we wish the ANN to emulate. Moreover, since the training data are taken from real-world measurements, they often contain random noise that complicates the task of learning the underlying patterns. There are two methods the ANN can use to minimize $\bar{C}$ over the training set: (1) learning the general characteristics of the process that produced the training data; (2) "memorizing" the exact training examples, including their associated noise. The latter process is known as *overfitting*[81], and it is a common pitfall in machine learning that results in

poor generalization to new data and extremely unstable behaviour.

In many ways, ANN overfitting is analogous to the more familiar polynomial overfitting shown in Figure 2.9. There, while the higher-order polynomial fit yields better performance at the discrete points that the fit was generated from, away from these points it strays from the underlying trend and occasionally explodes. The degree to which polynomial overfitting can be expected to occur is determined by the ratio of the number of points used for the fit to the degree of the polynomial function. A high ratio results in a much better general fit to the underlying trend, even though a low one may yield a lower fitting error.

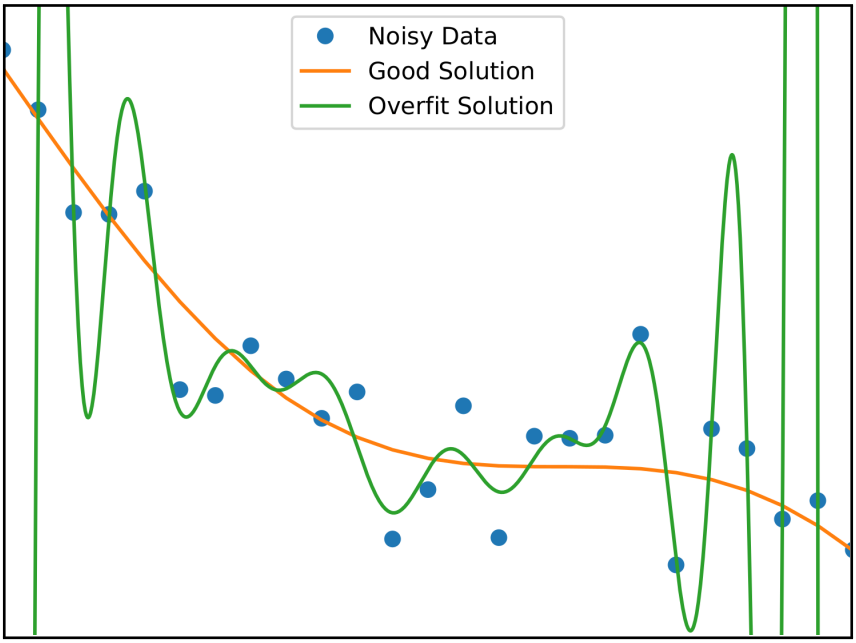

Figure 2.9: An illustrated example of polynomial overfitting, which in many ways is analogous to using overly complex ANNs on small training sets.

In the same way, the ratio of the number of free parameters in an ANN to the number of available training examples is critical. If it is too small, the network will learn spurious patterns that improve its training cost function but diminish its performance outside of the training set. As a general convention in machine learning, there should be at least ten times more examples in the

training set than there are trainable parameters in the ANN[82]. For a fully-connected ANN with N_{in} inputs, N_{HL} hidden layers with the i^{th} hidden layer having width w_i, and N_{out} outputs, the number of free parameters N_{fp} is given by:

$$N_{fp} = N_{in}w_1 + w_1w_2 + \ldots + w_{N_{HL}-1}w_{N_{HL}} + w_{N_{HL}}N_{out} \tag{2.37}$$

If biases are included for the hidden layers, this becomes

$$N_{fp} = N_{in}w_1 + (w_1+1)w_2 + \ldots + (w_{N_{HL}-1}+1)w_{N_{HL}} + (w_{N_{HL}}+1)N_{out} \tag{2.38}$$

An ANN with an input width of 10, two hidden layers of width 256 with biases, and an output width of 3 would have $10 \times 256 + 257 \times 256 + 257 * 3 = 69380$ trainable parameters. As ANNs go, this example is not particularly large, but already it requires at least 7×10^5 training examples to satisfy this convention. For many applications, including the one considered in this thesis, this size of training set is unattainable and alternative methods must be used to mitigate overfitting.

Collectively, these methods are known as *regularization*. Conventional methods for regularization include dropout networks, L1 regularization (also called lasso regression) and L2 regularization (also called ridge regression). In dropout networks (see Figure 2.10), a fraction of the neurons are randomly removed during training for one epoch at a time, forcing the network to better distribute learning tasks and preventing individual neurons from either (1) assuming responsibility for individual training examples; or (2) correcting the mistakes of other neurons[83]. More aggressive regularization can be performed by increasing the fraction of the neurons in each layer that are dropped out during each epoch.

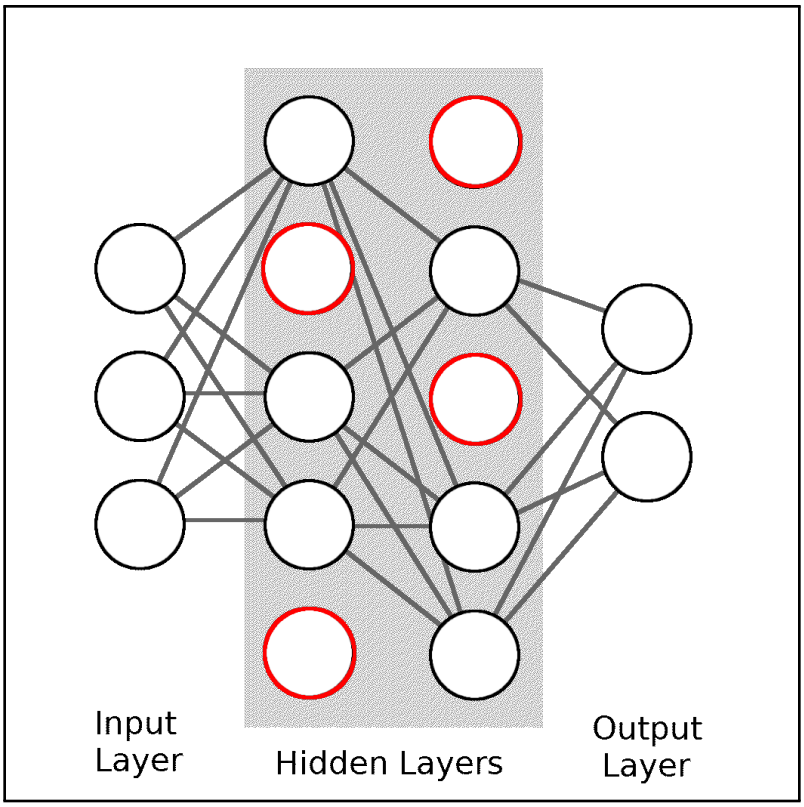

Figure 2.10: A dropout network with a dropout fraction of 0.4, meaning 2/5 neurons in each layer are randomly disconnected each training epoch. Disconnected neurons are shown in red.

In L1 regularization, an additional term is added to the overall cost function that penalizes the network for having too many non-zero weights:

$$\bar{C}_{L1} = \bar{C} + \lambda \sum_{i,j} |w_{i,j}| \tag{2.39}$$

Essentially, this term forces the network to find simpler representations of the training data even at the expense of training accuracy. The degree to which this occurs is controlled by the parameter λ.

In L2 regularization, a penalty term corresponding to the sum of squares of the internal weights is added:

$$\bar{C}_{L2} = \bar{C} + \lambda \sum_{i,j} (w_{i,j})^2 \tag{2.40}$$

As with dropout networks, this prevents any individual weight from growing too large during training, encouraging distribution of the learning throughout the network. Similar to L1 regularization the magnitude of regularization is controlled by a free parameter λ.

Earlier, an idealized illustration of the training process was introduced (Figure 2.7(a)), in which the average cost function $\bar{C}$ over the training set was smooth and convex with respect to any individual weight in the ANN. As mentioned, in practice this is usually not the case and the cost function landscape is more complicated. This is especially true when the ratio of training examples to free network parameters is small, since this introduces an element of chance as to whether specific combinations of weights fit well or poorly to the sparse training data. With larger training sets, this effect would tend to average out. This means that the cost function landscape may contain multiple local minima that can halt gradient descent, causing inconsistent training performance that depends on the random initialization of the weights (see Figure 2.11). Worse yet, these local minima likely correspond to poorer general performance than the network weight space that surrounds them.

A properly regularized network returns a sense of smoothness to the cost function landscape, since many of the overfit weight configurations that result in these local cost function minima and maxima are rendered inaccessible by the restrictions imposed by the regularization approach (e.g., small weights, sparse/distributed representations). As a result, there is less dependence of the solution on the random initial weight configuration (see Figure 2.12). Unfortunately, this typically comes at the cost of much slower training convergence.

In Chapter 3, I explore the use of a novel form of regularization called "super-convergence"[84] for tumour motion prediction. This regularization method entails drastically increasing the initial learning rate of the training process and cutting off training after fewer epochs have been performed. In

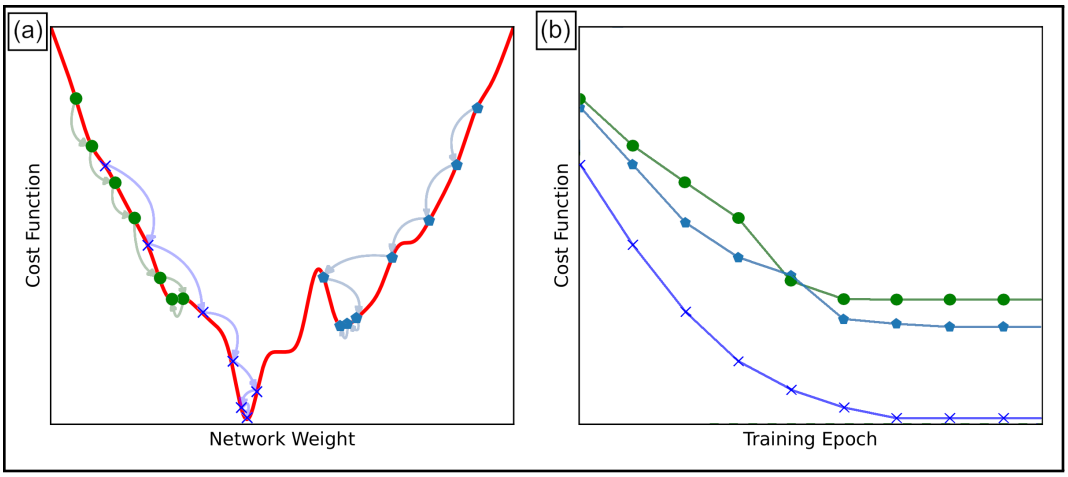

Figure 2.11: (a): An illustration of a more realistic cost function plot, showing the origin of initial weight dependency. Each network finds a different solution based on its random starting weights and learn rate. Some solutions are more overfit than others. (b) Another view of this process, showing how the cost function evolves over the training epochs. The network indicated with blue crosses achieves the best training loss, but overfitting may prevent it from being the best general solution.

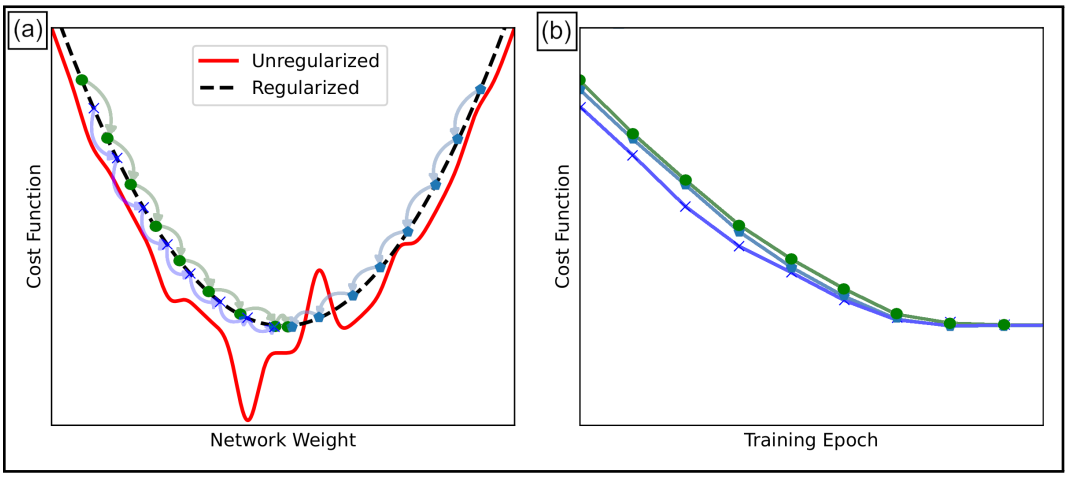

Figure 2.12: (a) An illustration of how regularization through dropout, L1 or L2 may affect the cost function terrain. Networks initialized identically to those in Figure 2.11 now converge to similar solutions corresponding to a single global minimum. Overfit solutions (deep minima on the solid line) are no longer accessible by the networks because of the constraints imposed by the regularization process. (b) Another view of this effect, showing how the cost function evolves over the training epochs.

contrast to conventional regularization, this actually accelerates the training process, which has many downstream benefits in terms of the practicality of

ANN-based tumour motion prediction.

The specific mechanism through which super-convergence regularizes the problem is still not well-described. To my understanding, however, the aggressive learning rate allows the network to evade or even escape local minima in the unregularized cost function landscape, and cutting off training early prevents it from settling into a local minimum as its step size decreases (see Figure 2.13). The net result is unprecedentedly fast convergence and a reduced (but non-zero) likelihood of overfitting.

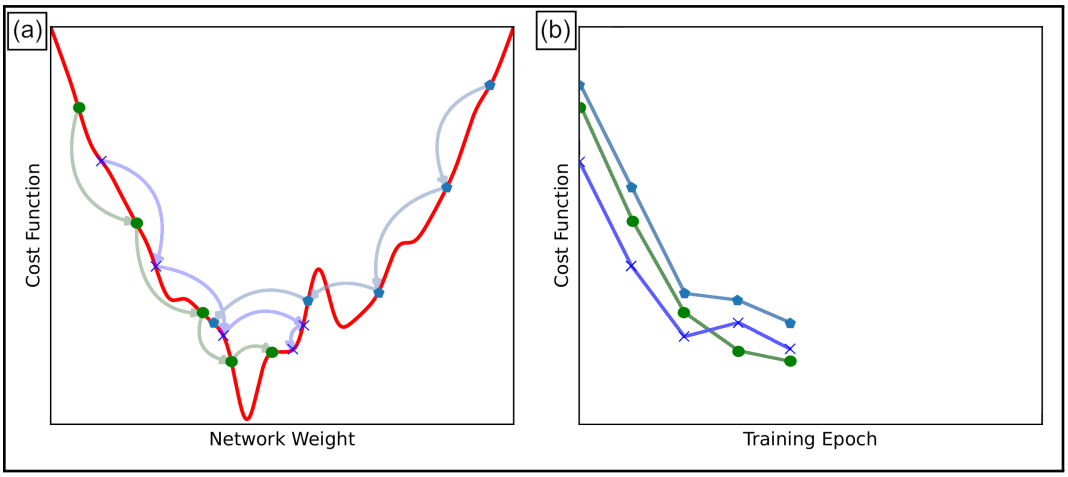

Figure 2.13: (a): An illustration of my interpretation of super-convergence regularization, again showing the same randomly initialized networks from the previous two figures. Aggressive training coupled with early halting of the training process helps to prevent overfitting (settling into a local minimum on the solid line). (b): Another view of this process illustrating the cost function as a function of training epoch, demonstrating the reduced training times compared to other regularization methods, but also the lingering dependence on random initial weights.

2.1.6 Recurrent Neural Networks

So far, the discussion in this chapter has assumed what is known as a feed-forward ANN. That is, the connections in the networks only exist between adjacent layers and always point in the direction of the output layer. Another way of looking at this is that each individual neuron takes in all of the outputs

of the previous layer simultaneously, then sends its output to all of the neurons in the subsequent layer simultaneously, with no formal sense of order. This has been done largely for the sake of simplicity.

However, for some problems such as language processing and time series forecasting, the order of the inputs can be as important as their values. RNNs were developed specifically to handle sequential data. For these networks, inputs are considered one at a time, and self-connections allow for each input to modify one or more internal states (called hidden states) of the neuron before it processes the next data point. The hidden state can be modified to store contextual information that might be helpful toward understanding subsequent data points. Figures 2.14(a) and (b) illustrate the general concept of an RNN, showing a neuron in both its *folded* and *unfolded* forms.

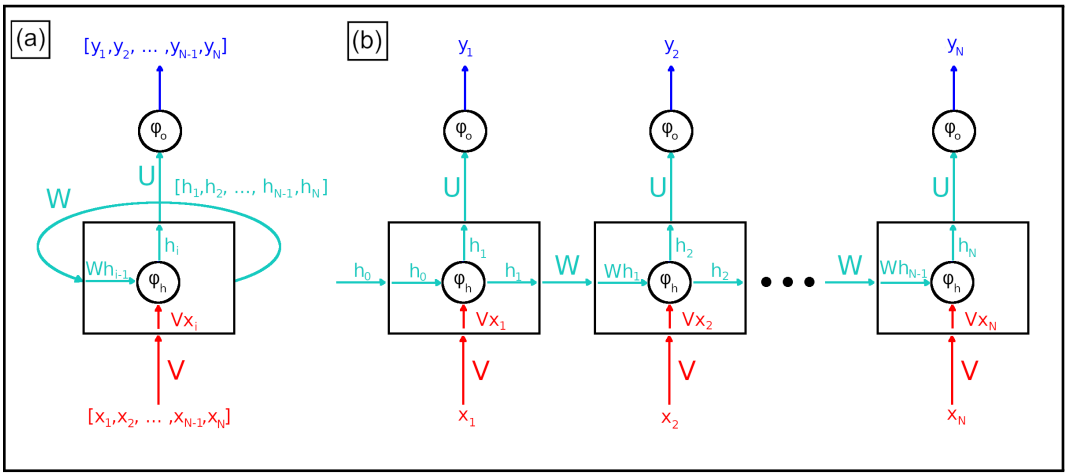

Figure 2.14: (a): A folded view of a recurrent neural network. (b): The same network, but unfolded to show its temporal structure.

Mathematically, if the N sequential inputs are written as $[x_1, x_2, \ldots, x_{N-1}, x_N]$, the neuron will have $N+1$ hidden states $[h_0, h_1, \ldots, h_{N-1}, h_N]$. These states are connected through a weight matrix $\mathbf{U}$ to an output activation function ϕ_o to generate N outputs $[y_1, y_2, \ldots, y_{N-1}, y_N]$. The hidden state has its own internal activation function ϕ_h that acts on the previous hidden state with a weight matrix $\mathbf{W}$ and the input from the current time step with a weight

matrix $\mathbf{V}$. Together,

$$h_i = \phi_h(\mathbf{V}x_i + \mathbf{W}h_{i-1}), \qquad y_i = \phi_o(\mathbf{U}h_i) \tag{2.41}$$

Since both $\vec{X}$ and $\vec{Y}$ are N-vectors representing inputs and outputs at discrete time points, this is known as a sequence-to-sequence transformation. However, as is the case in this thesis, what may be desired is a transformation from a sequence to a single future value. In this case, only the last output y_N can be considered when training the network – the previous outputs need not be calculated, even during training. For completeness, however, for the remainder of this section I will continue to assume the more general sequence-to-sequence transformation.

Initially, h_0 is randomly assigned, as are the weight matrices $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{W}$. These values are then all trained through cycles of forward- and backpropagation, with the aim of minimizing the average of the cost functions C_i, which are then averaged over all time points. However, because each hidden state h_i is influenced by the previous hidden states and inputs, this backpropagation not only needs to take place through the layers of the network, but also through time. For simplicity, assuming that $N = 2$:

$$\begin{gathered} y_1 = \phi_o(\mathbf{U}h_1) = \phi_o(\mathbf{U}\phi_h(\mathbf{V}x_1 + \mathbf{W}h_0)) \\ y_2 = \phi_o(\mathbf{U}h_2) = \phi_o(\mathbf{U}\phi_h(\mathbf{V}x_2 + \mathbf{W}h_1)) = \\ \phi_o(\mathbf{U}\phi_h(\mathbf{V}x_2 + \mathbf{W}(\phi_h(\mathbf{V}x_1 + \mathbf{W}h_0)))) \end{gathered} \tag{2.42}$$

During backpropagation, the partial derivative of each cost function with respect to $\mathbf{U}$, $\mathbf{V}$, $\mathbf{W}$ and h_0 needs to be calculated. For C_1 generated by y_1:

$$\frac{\partial C_1}{\partial \mathbf{U}} = \frac{\partial C_1}{\partial y_1}\frac{\partial y_1}{\partial \mathbf{U}} \tag{2.43}$$

where, as before, these terms are made easy to calculate by selecting an appropriate C and ϕ_o. For $\mathbf{V}$, $\mathbf{W}$ and h_0,

$$\begin{aligned}
\frac{\partial C_1}{\partial \mathbf{V}} &= \frac{\partial C_1}{\partial y_1}\frac{\partial y_1}{\partial h_1}\frac{\partial h_1}{\partial \mathbf{V}} \\
\frac{\partial C_1}{\partial \mathbf{W}} &= \frac{\partial C_1}{\partial y_1}\frac{\partial y_1}{\partial h_1}\frac{\partial h_1}{\partial \mathbf{W}} \\
\frac{\partial C_1}{\partial h_0} &= \frac{\partial C_1}{\partial y_1}\frac{\partial y_1}{\partial h_1}\frac{\partial h_1}{\partial h_0}
\end{aligned} \tag{2.44}$$

Again, the second term in each expression is related to the gradient of the output activation function ϕ_o and the third is related to the gradient of the hidden activation function ϕ_h at points that were evaluated in the forward propagation step.

For $i = 2$, things get a little more complicated:

$$\frac{\partial C_2}{\partial \mathbf{U}} = \frac{\partial C_2}{\partial y_2}\frac{\partial y_2}{\partial \mathbf{U}} \tag{2.45}$$

is similar to the case of $i = 1$, but

$$\begin{aligned}
\frac{\partial C_2}{\partial \mathbf{V}} &= \frac{\partial C_2}{\partial y_2}\frac{\partial y_2}{\partial h_2}\frac{\partial h_2}{\partial h_1}\frac{\partial h_1}{\partial \mathbf{V}} \\
\frac{\partial C_2}{\partial \mathbf{W}} &= \frac{\partial C_2}{\partial y_2}\frac{\partial y_2}{\partial h_2}\frac{\partial h_2}{\partial h_1}\frac{\partial h_1}{\partial \mathbf{W}} \\
\frac{\partial C_2}{\partial h_0} &= \frac{\partial C_2}{\partial y_2}\frac{\partial y_2}{\partial h_2}\frac{\partial h_2}{\partial h_1}\frac{\partial h_1}{\partial h_0}
\end{aligned} \tag{2.46}$$

contains the product of gradients of ϕ_h evaluated at two different points. More generally, for the n^{th} output of an RNN with an arbitrarily large input size $N > n$,

$$\begin{aligned}
\frac{\partial C_n}{\partial \mathbf{U}} &= \frac{\partial C_n}{\partial y_n}\frac{\partial y_n}{\partial \mathbf{U}} \\
\frac{\partial C_n}{\partial \mathbf{V}} &= \frac{\partial C_n}{\partial y_n}\frac{\partial y_n}{\partial h_n}\left(\prod_{j=2}^{n}\frac{\partial h_j}{\partial h_{j-1}}\right)\frac{\partial h_1}{\partial \mathbf{V}} \\
\frac{\partial C_n}{\partial \mathbf{W}} &= \frac{\partial C_2}{\partial y_2}\frac{\partial y_2}{\partial h_2}\left(\prod_{j=2}^{n}\frac{\partial h_j}{\partial h_{j-1}}\right)\frac{\partial h_1}{\partial \mathbf{W}} \\
\frac{\partial C_n}{\partial h_0} &= \frac{\partial C_2}{\partial y_2}\frac{\partial y_2}{\partial h_2}\left(\prod_{j=2}^{n}\frac{\partial h_j}{\partial h_{j-1}}\right)\frac{\partial h_1}{\partial h_0}
\end{aligned} \tag{2.47}$$

These repeated multiplications of hidden state partial derivatives (each of which is itself composed of the product of a weight matrix and the derivative of the hidden state activation function) can be problematic. If each term is small (usually the fault of initializing to a flat region of the activation function), the gradient could prevent any meaningful updates from being applied to the weights and initial hidden state, and if each term is large (usually the fault of inappropriately large weights working their way into the weight matrix) the gradient could rapidly grow and the training process could become unstable or even divergent. These are called the vanishing and exploding gradient problems, respectively. They are well-known for deep feed-forward ANNs, and because of the stacking of products that occurs in RNNs with long input sequences, they reappear here. These problems can be avoided by simply truncating the product of the gradients, but this can preclude the learning of longer-term dependencies.

LSTM-RNNs[85] were introduced to address this shortcoming of traditional RNNs. Their neurons have a much more complicated internal structure (see Figure 2.15) – in addition to the hidden state, LSTM-RNNs have an additional state called the cell state, $[c_0, c_1, \ldots, c_{N-1}, c_N]$. Four internal gates exist within each neuron, which can act on either the cell state or the hidden state:

- The *forget* gate, which determines how much of each component of the cell state to keep based on the new inputs and the previous hidden state
- The *input* gate, which parses new inputs and determines how much of them to incorporate into the new cell state
- The *input modulation* gate, which converts the instructions of the input gate into a form that leads to faster convergence
- The *output* gate, which queries the inputs and cell state to determine

the next hidden state

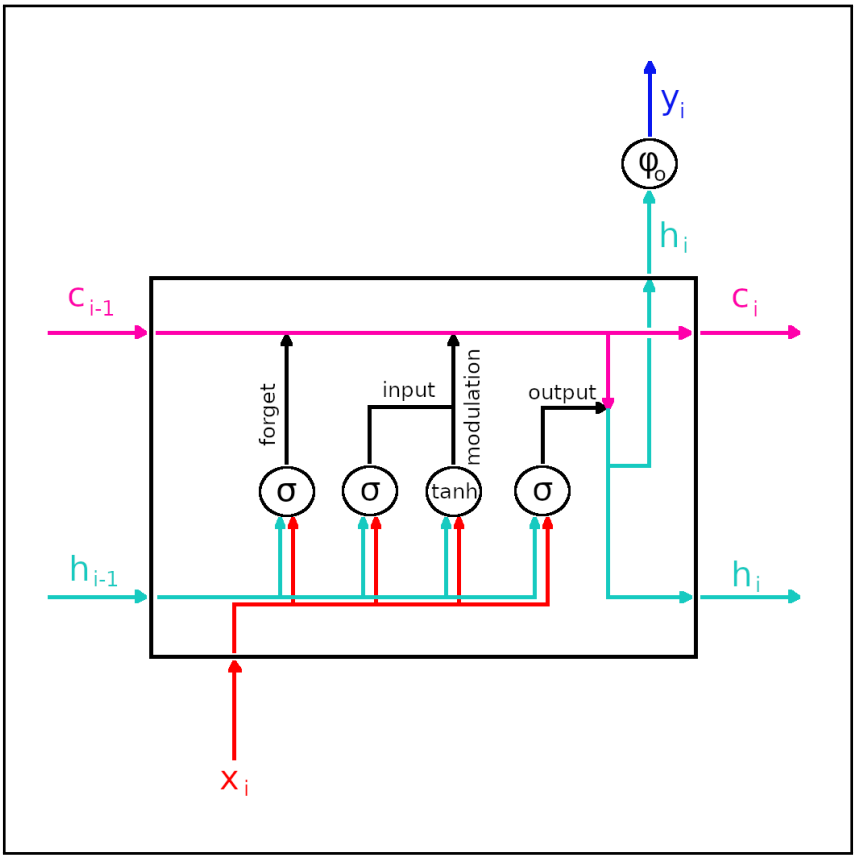

Figure 2.15: An illustration of a LSTM-RNN neuron and its internal gates. $\mathbf{W}$ is a multi-dimensional matrix that contains $\mathbf{W}_f$, $\mathbf{W}_i$, $\mathbf{W}_o$ and $\mathbf{W}_c$.

Each of these internal gates has their own associated weight matrix. If f_i, i_i, and o_i are the values of the forget, input, and output gates at time point i, h_i and c_i are the hidden and cell states at that time point, and $\mathbf{W}_f$, $\mathbf{W}_i$, $\mathbf{W}_c$ and $\mathbf{W}_o$ are the weight matrices for the forget, input, input modulation and output steps respectively, then the mathematical operation of the LSTM-RNN cell is given by:

$$
\begin{aligned}
f_i &= \sigma(\mathbf{W}_f[x_i, h_{i-1}]) \\
i_i &= \sigma(\mathbf{W}_i[x_i, h_{i-1}]) \\
o_i &= \sigma(\mathbf{W}_o[x_i, h_{i-1}]) \\
c_i &= f_i * c_{i-1} + i_i * \tanh(\mathbf{W}_c[x_i, h_{i-1}]) \\
h_i &= o_i * \tanh(c_i) \\
y_i &= \phi_o(\mathbf{U}h_i)
\end{aligned}
\tag{2.48}
$$

Here, σ is the sigmoid function, tanh is the hyperbolic tangent function, and $\mathbf{U}$ and ϕ_o are the weight matrix and activation functions that connect the hidden state to the output, as before. The internal sigmoid and tanh activation functions are typically left unchanged, while the output activation function ϕ_o can be varied. Essentially, this form of connection allows the network to "store" inputs in the cell state, where they can more directly affect the later functioning of the neuron without the risks associated with being located behind multiple gradients.

LSTM-RNNs require considerably more calculations to be performed during both forward and backpropagation compared to a feed-forward network with a similar architecture, but their ability to "comprehend" the temporal relationships between their inputs yields improvements in performance that are usually deemed well worth their computational cost[86].

2.1.7 Network Adaptation

Recursion in neural networks helps them to identify and act on short-term patterns in sequential data. However, these RNNs will only remain accurate as long as those patterns remain relevant. As mentioned in Chapter 1, respiratory motion patterns can change considerably, even on intrafractional time scales. Maintaining accuracy in tumour motion prediction therefore requires a method for updating an RNN's training throughout a treatment to keep pace with

changes in respiratory patterns. This process is called network adaptation.

In MR-based intrafractional tumour tracking, new tumour positions are being acquired at a minimum rate of 4 frames per second. Since it takes considerably more time than this to train an RNN, there are two options for adaptation strategies:

- Perform an abbreviated ($<$ 250 ms) training cycle, updating network weights before the next prediction is due
- Perform a full training cycle, using an old network to make predictions until the new one is ready

Typically under the first approach, one cycle of forward- and backpropagation is performed using only the most recent data point as the training "set" in order to limit the amount of time required for adaptation. In some cases, the learning rate for the adaptation is fixed, while in others an optimizer continually manages the learning rate during adaptation, passing it between steps. The latter approach effectively allows the optimizer to adjust the aggressiveness of adaptation based on how quickly the respiratory patterns are changing.

This strategy is called online learning, and while it has been generally successful in the past it has a few limitations. First, the initial training set might be reinforced over hundreds, thousands or even hundreds of thousands of epochs while new data are only learned once. This could result in the network retaining too much memory of respiratory patterns that have lost their relevance. Second, there may be difficulty in predicting an appropriate fixed learning rate for adaptation. If an optimizer is being used, then the learning rate inherited from initial training might be inappropriate for adaptation, and it may therefore take the optimizer some time to adapt to its new task. Finally,

transient changes in respiratory patterns (a cough, for example) may destabilize the network, since their large associated errors could result in drastic changes to the network weights.

The second strategy avoids these pitfalls of online learning, since every data point is learned equivalently, and any transient changes in respiratory patterns can be buffered by the rest of the training set. However, under this approach, the fastest that a network can adapt to changes in respiratory patterns is equal to the amount of time required for a full training cycle. Historically for respiration-induced tumour motion prediction, this has been somewhere in the range of tens of minutes to several hours, which is far too long to adequately capture intrafractional variation in respiratory patterns. In Chapter 3, I introduce a method for reducing network training times down to a few seconds, making this second approach feasible. I compare it to conventional online learning, and show that it results in better adaptation when presented with shifting respiratory patterns.

2.2 Magnetic Resonance Imaging

For DTTRT, MRI offers the distinct advantages of (1) unparalleled soft tissue contrast, (2) markerless visualization of both the tumour position and shape, and (3) no additional patient dose due to imaging. However, it is a relatively slow imaging technique compared to external surrogate tracking or fluoroscopy. As mentioned previously, this is why MRI-based dynamic tumour-tracked radiotherapy requires a non-linear prediction method, while hardware like the CyberKnife or VERO systems can make use of simple linear predictors.

In this section, I will briefly introduce the basics of MRI, starting with the fundamental physics and a description of spatial encoding for image formation. An exhaustive introduction to the theory and practical application of MRI is beyond the scope of this thesis. Rather, I will be focusing on the basic factors

that determine MRI acquisition speed (and therefore the system delay for dynamic tumour-tracked radiotherapy using MR tracking).

2.2.1 Fundamental Physics

The constituent particles of atomic nuclei, protons and neutrons, are both ground-state baryons, meaning they have a spin (a quantized intrinsic angular momentum) of $\frac{1}{2}$. A nucleus itself can therefore have either an integer or half-integer spin depending on how many protons and neutrons it contains – if both counts are even, the nucleus will have a spin of zero, if one is odd and the other even it will have half-integral spin, and if both are odd it will have integral spin.

The human body primarily consists of water, about 50%-60% by mass. A water molecule is composed of one oxygen atom and two hydrogen atoms. The most abundant isotope of oxygen, ^{16}O, has 8 protons and 8 neutrons and is therefore spin-0, but the most abundant isotope of hydrogen, ^{1}H, consists of a single proton and therefore has a half-integer nuclear spin. Specifically, in the ground state its spin quantum number S can be either $+\frac{1}{2}$ or $-\frac{1}{2}$.

Any particle with a non-zero spin will interact with an external magnetic field. The strength of this coupling depends on its nuclear magnetic moment $\vec{\mu}$, which is proportional to its intrinsic spin angular momentum $\vec{S}$:

$$\vec{\mu} = \gamma \vec{S} \tag{2.49}$$

Here, the proportionality constant γ is called the gyromagnetic ratio of the particle. ^{1}H has a relatively large gyromagnetic ratio compared to other common nuclei at 42.58 Hz/G.

When a uniform, static external magnetic field B_0 is present, the spin states of the ^{1}H nucleus will split into two energy levels based on their alignment with B_0 (called the Zeeman effect). These levels will be separated by an energy

$2\mu B_0$ with the spin state that is aligned with the external magnetic field at a lower energy than the anti-aligned spin state (assuming a positive γ, which is typically the case and is true for ^{1}H). A transition between these states would require the absorption or emission of a photon with frequency ω_0, such that:

$$\Delta E = 2\mu B_0 = \hbar\omega_0 \tag{2.50}$$

which means that

$$\omega_0 = \frac{2\mu B_0}{\hbar} \tag{2.51}$$

If $\vec{\mu}$ and $\vec{B_0}$ are not exactly aligned (or anti-aligned), the external field also creates a torque $\vec{\tau}$ given by

$$\vec{\tau} = \vec{\mu} \times \vec{B_0} \tag{2.52}$$

Since

$$\vec{\tau} = \frac{d\vec{S}}{dt} = \frac{1}{\gamma}\frac{d\vec{\mu}}{dt} \tag{2.53}$$

then

$$\frac{d\vec{\mu}}{dt} = \gamma\vec{\mu} \times \vec{B_0} \tag{2.54}$$

This equation describes oscillatory behaviour with a frequency γB_0 (called the Larmor frequency), namely a precession of $\vec{\mu}$ about $\vec{B_0}$. A quantum mechanical treatment of this phenomenon, which is beyond the scope of this thesis, demonstrates that this Larmor frequency is identical to ω_0 from Equation 2.51:

$$\omega_0 = \gamma B_0 \tag{2.55}$$

At thermal equilibrium, the relative populations of the different spin orientations can be obtained through Maxwell-Boltzmann statistics:

$$\frac{p_+}{p_-} = \exp\left(-\frac{\Delta E}{k_B T}\right) \tag{2.56}$$

where p_+ and p_- represent the probabilities of finding a nucleus in the higher energy or lower energy state, respectively, $\Delta E > 0$ is the difference between the high and low energy states, k_B is the Boltzmann constant and T is the temperature of the system. ΔE is generally quite small relative to $k_B T$, so there will usually only be a small imbalance between the populations of aligned and anti-aligned states.

Consider a volume that contains only one type of nucleus with a non-zero nuclear magnetic moment. In the absence of an external magnetic field, the nuclear magnetic moments will be randomly oriented, and so they will tend to cancel each other out. The magnetic moment of the volume as a whole will be zero (Figure 2.16(a)). In the presence of an external magnetic field, the spins will precess around the external field lines, with a slight preference for an aligned state rather than an anti-aligned one. This will create a net magnetic moment M_0 for the whole volume that is proportional to the number of spins contained within that volume (Figure 2.16(b)).

If a pulsed, single-frequency magnetic field $\vec{B_1}(t)$ tuned to the Larmor frequency and perpendicular to $\vec{B_0}$ is now applied to the volume, M_0 will experience a torque that will tip it away from $\vec{B_0}$, causing the net magnetization to precess about $\vec{B_0}$ at the Larmor frequency, just as the individual nuclear magnetic moments did earlier (see Figure 2.17(a)). This constructive precession, called nuclear magnetic resonance, results in a time-varying magnetic flux that can be detected with an appropriately oriented induction coil tuned to the Larmor frequency. The amplitude of the current induced in the coil will be proportional to the magnitude of M_0, which itself is proportional to

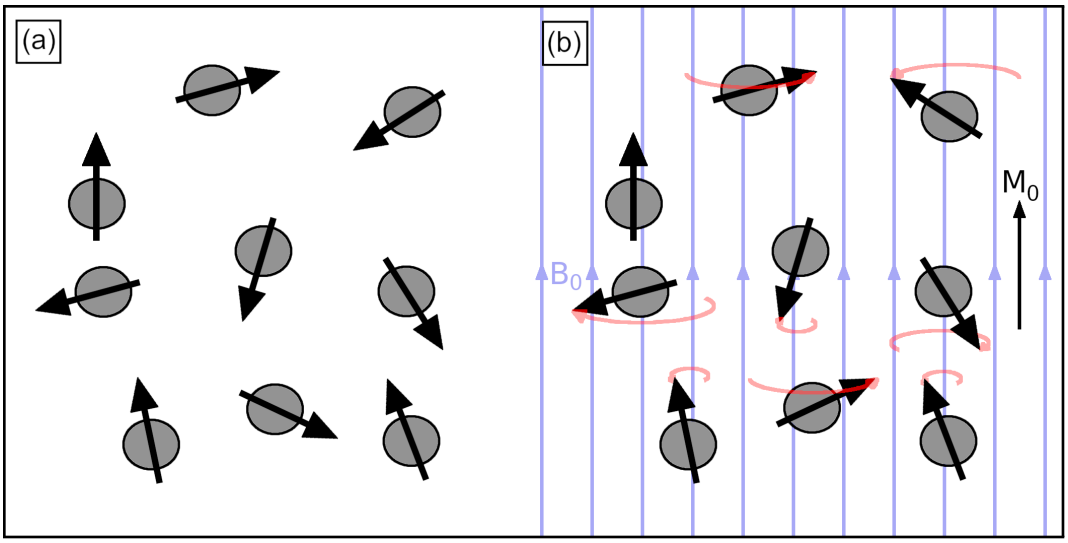

Figure 2.16: (a): Nuclear spins are randomly oriented in the absence of an external magnetic field, resulting in zero net magnetization. (b): Application of a uniform external magnetic field, such as the main field of an MRI, results in a slight favouring of one spin state, generating a net magnetization. It also causes incoherent precession of the spins.

the number of nuclei in the volume being measured. A $\vec{B_1}(t)$ of sufficient magnitude and duration will tip M_0 until it is completely perpendicular to $\vec{B_0}$, maximizing the intensity of the signal that is emitted (Figure 2.17(b)).

Nuclear magnetic resonance allows for the concentration of specific nuclei within a sample to be characterized, and because the Larmor frequency of a nucleus is affected by its chemical environment (with frequency shifts on the order of a few parts per million), chemical analysis of a substance with an active nucleus is also possible. However, in order to generate an image of an object, a method for coupling the Larmor frequency to the spatial location of the nucleus is required.

2.2.2 Spatial Encoding and Image Formation

When the $\vec{B_1}(t)$ pulse is applied, it will be absorbed by any nucleus having a Larmor frequency that matches the frequency of the pulse. If the only magnetic field present is the static, uniform $\vec{B_0}$, that will include all of the

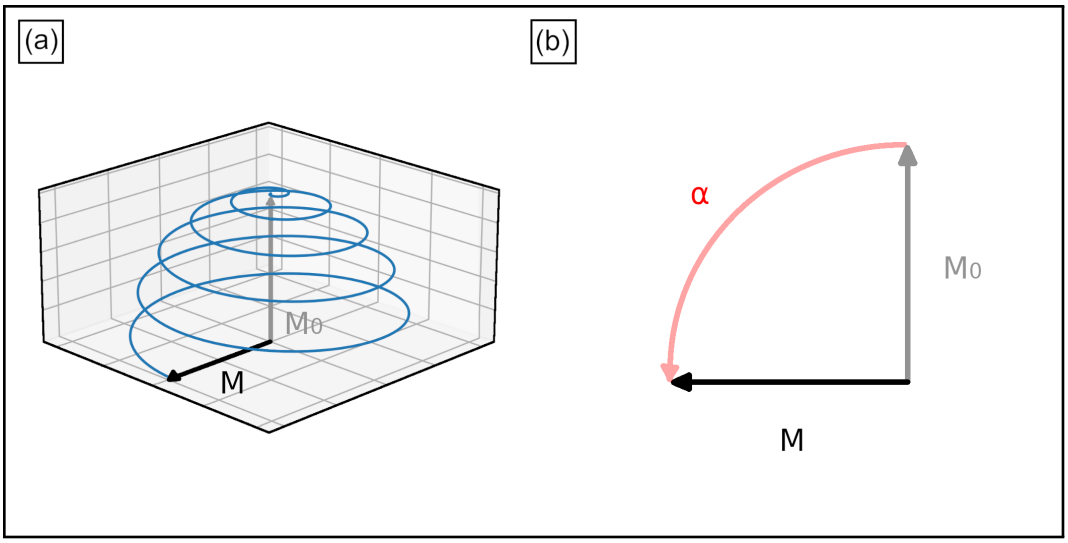

Figure 2.17: (a): The excitation pulse causes the net magnetization of the volume to simultaneously precess and tip away from the z direction, resulting in a path like that shown in blue. (b) The motion of the magnetization is much simpler in a reference frame co-rotating at the Larmor frequency. The tipping or nutation angle α is determined by the strength and duration of the pulse.

nuclei of interest in the sample, regardless of their position. If $\vec{G}(\vec{x})$ is a spatially-varying field that is parallel to $\vec{B}_0$, then the Larmor frequency within the sample will itself become spatially dependent:

$$\omega(\vec{x}) = \gamma|\vec{B}_0 + \vec{G}(\vec{x})| \tag{2.57}$$

$\vec{G}(\vec{x})$ is called a gradient field, because its typical form is a constant spatial gradient across the sample in one dimension (see Figure 2.18). If $\vec{B}_0$ is taken to be in the z-direction, as is customary, and $\vec{G}_x$ is a field oriented parallel to B_0 with a gradient along the x-direction such that

$$\vec{G}_x = G_x x\hat{z} \tag{2.58}$$

then Equation 2.57 becomes

$$\omega(x) = \gamma B_0 + \gamma G_x x = \omega_0 + \gamma G_x x \tag{2.59}$$

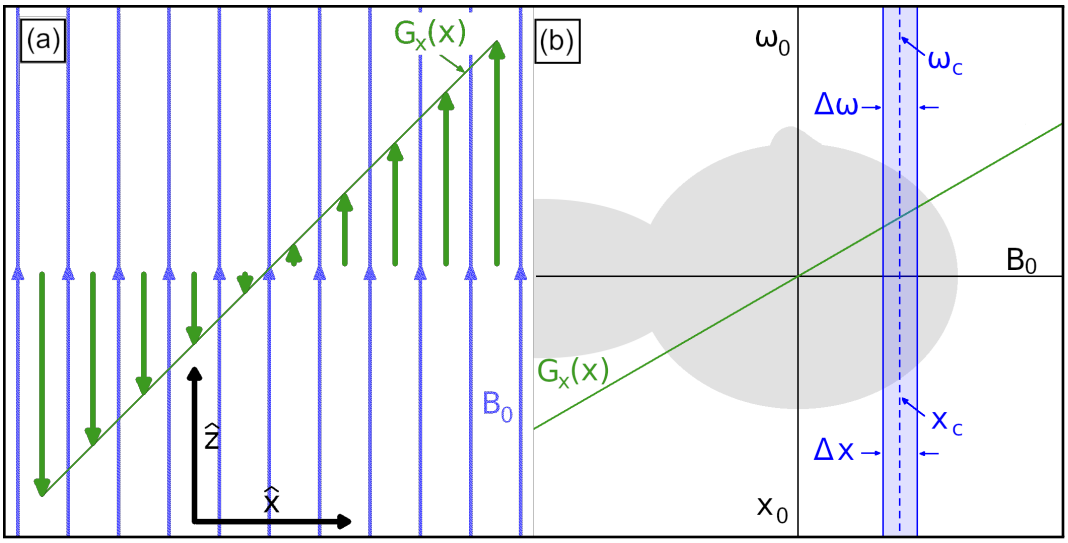

Figure 2.18: (a): A gradient field that varies along the x-direction and is oriented in the z-direction, parallel to B_0. (b): This gradient creates an x-dependence of the Larmor frequency, allowing for the selective excitation of a single slice of nuclei in a sample. The slice width is related to the bandwidth of the excitation pulse.

Next, if the single-frequency $\vec{B_1}(t)$ pulse is replaced with one with a bandwidth $\Delta\omega$ centred at ω_c, any nuclei with a Larmor frequency between $\omega_c - \Delta\omega/2$ and $\omega_c + \Delta\omega/2$ will be excited by this pulse. This corresponds to a slab of the volume along the x-direction with thickness Δx and center x_c such that

$$x_c = \frac{\omega_c - \omega_0}{\gamma G_x} \tag{2.60}$$

and

$$\Delta x = \frac{\Delta\omega}{\gamma G_x} \tag{2.61}$$

This process is known as slice-selection, and it allows for a spatially localized excitation of nuclear spins, limiting the volume that will eventually emit signal to a slice with a finite thickness Δx.

A similar concept can be applied when reading out the signal from the excited slice. Applying a gradient $\vec{G}_y = G_y y\hat{z}$ that varies along the y-direction

during readout, the net magnetization along the y-direction will precess with different frequencies depending on the position of the nuclei in space. The readout coil signal can then be then sampled at finite time points, and a discrete Fourier transform can be used to extract discrete frequency bins from the measured temporal signal. Since G_y ties the precession frequency of a nucleus to its position in space, this frequency signal equates to a spatial mapping of the nuclei in the sample along the y-direction. This process is known as frequency encoding, and is shown in Figure 2.19.

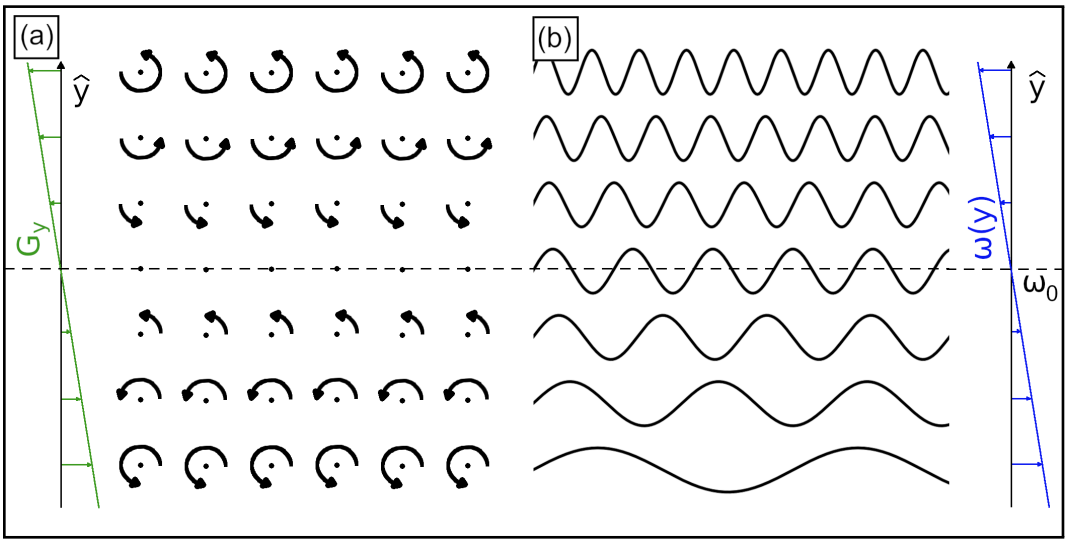

Figure 2.19: A gradient field varying in the y-direction, tying the Larmor frequency to the y-coordinate of the nuclei. The rotations shown in the left panel illustrate the varying angular frequency relative to the Larmor frequency at a field strength of B_0.

Finally, if a third gradient field $\vec{G}_z = G_z z\hat{z}$ that varies in the z-direction is briefly applied prior to the readout for a time Δt, then the nuclei experiencing that field will briefly precess with a different frequency than a nucleus experiencing no gradient. They will then acquire a spatially-varying phase $\Delta\phi(z)$ with:

$$\Delta\phi(z) = \Delta\omega(z)\Delta t = \gamma G_z z \Delta t \tag{2.62}$$

By varying the strength of the applied field G_z and keeping its duration Δt constant, several readouts with different phase shift magnitudes can be acquired. This essentially causes the points experiencing a different G_z to oscillate at different frequencies across measurements, since the phase shifts they accumulate grow at different rates. Again, after a Fourier transform of this phase signal into discrete frequency bins, it can then be mapped into a spatial variation of signal in real space. This process is known as phase encoding, and it is illustrated in Figure 2.20.

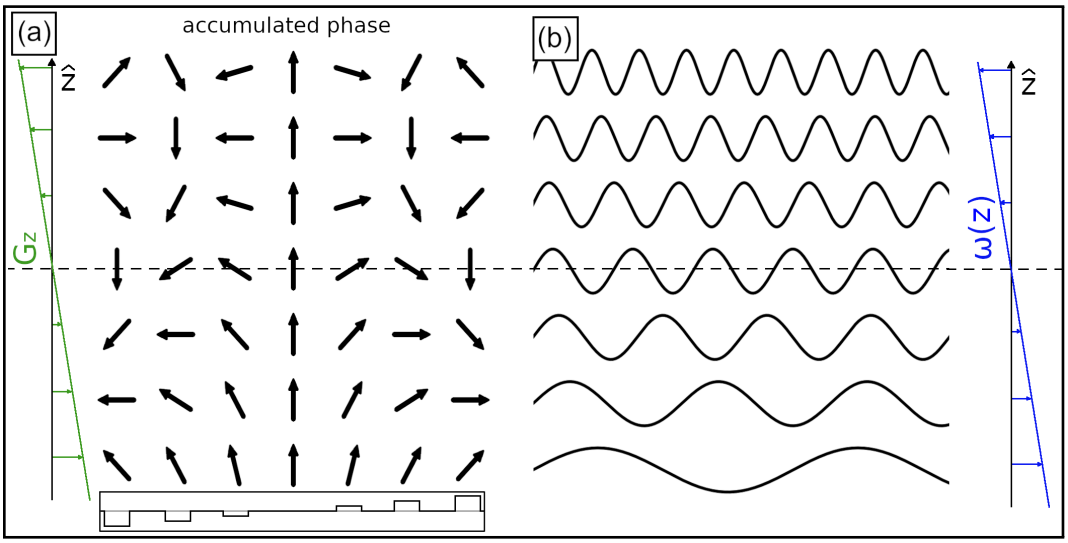

Figure 2.20: An illustration of phase encoding, wherein applying gradient fields of different strengths (inset, bottom of left panel) in the z-direction result in oscillatory behaviour with a spatially-varying frequency.

Figure 2.21 shows a possible sequence of magnetic field pulses that could be used to create a 2D image. It involves first applying an excitation pulse B_1 with a slice-select gradient G_x active to define a slice of interest in the x-direction, then applying a brief phase-encoding gradient G_z to the sample to select the point in phase space that is to be measured, then acquiring the signal from the sample with a frequency-encoding gradient G_y active. This process would then repeated for all of the points that are intended to be sampled in phase space, and then for each slice that is intended to be imaged in real space.

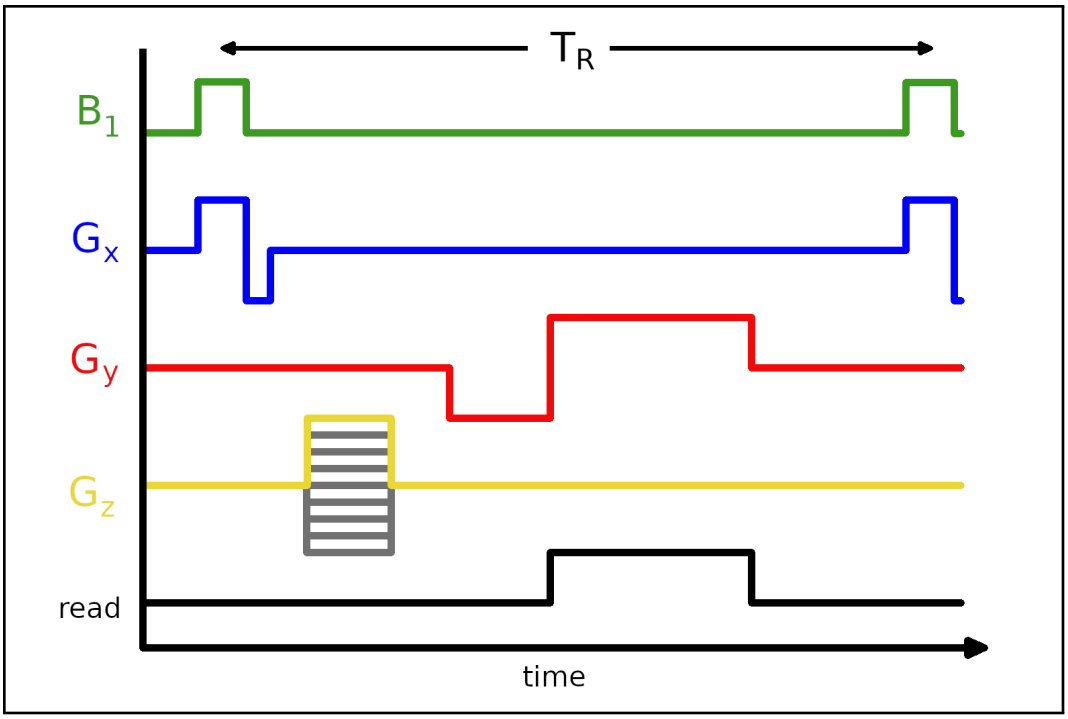

Figure 2.21: A basic MRI pulse sequence, consisting of an excitation pulse concurrent with a slice-select gradient, followed by a phase-encoding gradient, followed by a frequency-encoding gradient during readout. The repetition rate of the sequence, T_R, depends on how fast the initial net magnetization can recover.

Within each slice, it is convenient to imagine this sequence of events as traversing a discrete trajectory through two-dimensional k-space (see Figure 2.22). In this space, a frequency-encoding gradient will cause a translation along the x-axis, and a phase-encode gradient will result in motion along the y-axis. The number of grid points in the frequency-encoded direction will be determined by the number of temporal samples taken of the continuous RF signal during readout, and the number of grid points in the phase-encoded direction will depend on the number of different phase-encode gradient strengths used to obtain the image. The extent of k-space in both dimensions depends on the maximum strength of the applied gradients. From the properties of the Fourier transform, this "field of view" in k-space inversely correlates to the pixel spacing in real space, while the k-point spacing inversely correlates to

the field of view in real space.

The low spatial frequency (i.e., contrast) information is stored near the origin of k-space, while the periphery contains information about the high spatial frequency structures (i.e., the finer detail). This means that the gross position of the tumour in the image will primarily depend on where it was located when the central region of k-space was sampled. The acquisition-related component of the system delay is given by the amount of time that elapses between the most recent localization of the tumour and the beginning of image reconstruction. For a symmetric k-space trajectory, the tumour is localized in the middle of the acquisition and reconstruction begins at the end, so the resulting delay should be approximately half of the total acquisition time. For a 4 Hz acquisition rate, this would imply an acquisition-related system delay component of 125 ms.

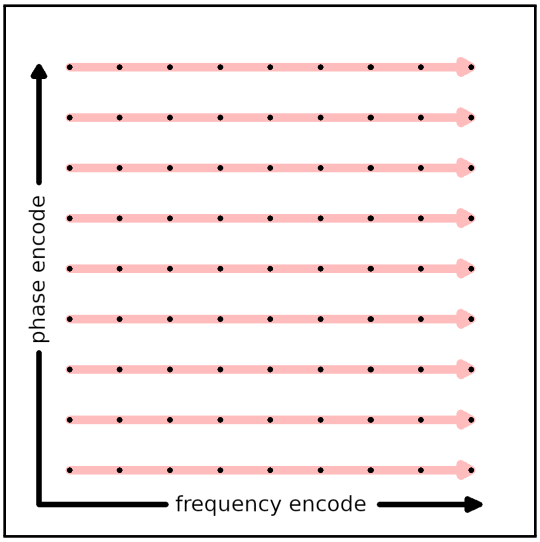

Figure 2.22: An illustration of the path taken through k-space for the pulse sequence shown in Figure 2.21.

There exists a limit to how closely these excitation pulses can be spaced together, since the component of the net magnetization parallel to $\vec{B}_0$ needs to adequately recover between excitations. For the pulse sequence shown in Figure 2.21, if T_R is the length of time between consecutive pulses that will

allow for sufficient recovery of the equilibrium magnetization, and N_p is the number of sampling points in phase space, then the total time required to obtain a 2D slice is $t_{acq,2D} = T_R N_p$. If a volumetric image is composed of a stack of N_{slice} slices, then the acquisition time for a 3D image will be given by $t_{acq,3D} = T_R N_p N_{slice}$.

T_R is typically on the order of hundreds to thousands of ms, and N_p is also typically > 100 to obtain images with an appropriate field of view and spatial resolution. A single slice under this approach would therefore take tens to hundreds of seconds to acquire, which is far too slow to capture respiratory motion in real-time.

2.2.3 Image Acceleration Techniques

One solution to this problem is called echo-planar imaging (EPI), wherein multiple lines of k-space are acquired per excitation pulse. A simple illustration of a pulse sequence that would allow this is shown in Figure 2.23. This equates to taking a "zig-zag" path through k-space, either after a single excitation pulse (single-shot EPI) or over the course of several (but $< N_p$) excitations (multi-shot EPI). A similar technique is known as turbo spin echo (TSE) imaging, with the difference between EPI and TSE being related to their mechanisms for rewinding the dephasing of spins due to B_0 inhomogeneities (EPI is a gradient-recalled echo sequence, while TSE is a spin-echo sequence).

Steady-state free precession (SSFP)[87] techniques involve repeated excitations with opposing flip angles (i.e., α and $-\alpha$), eventually establishing a near-constant magnitude of signal-generating transverse magnetization. In balanced SSFP (bSSFP, see Figure 2.24), the net gradient in each direction is zero when integrated over a full TR, allowing for rephasing of the nuclei between excitations. As a result, phase-encode lines in k-space can be rapidly filled, typically at a rate of a few ms per TR with a relatively high signal-to-

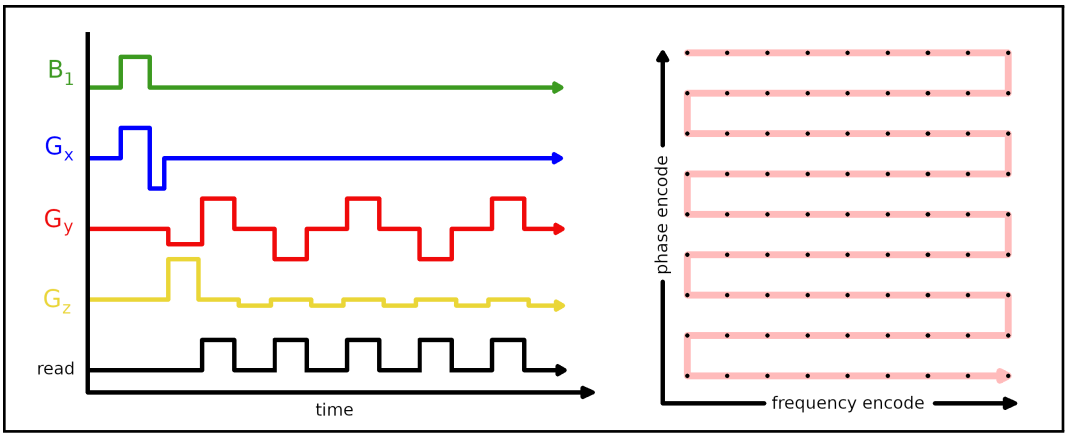

Figure 2.23: A simple illustration of a single-shot EPI pulse sequence, and the resulting path through k-space.

noise ratio for a SSFP approach[88]. However, one of the major drawbacks of this method is the appearance of "banding" artefacts related to inhomogeneities in the B_0 field, which can obfuscate important anatomy[88].

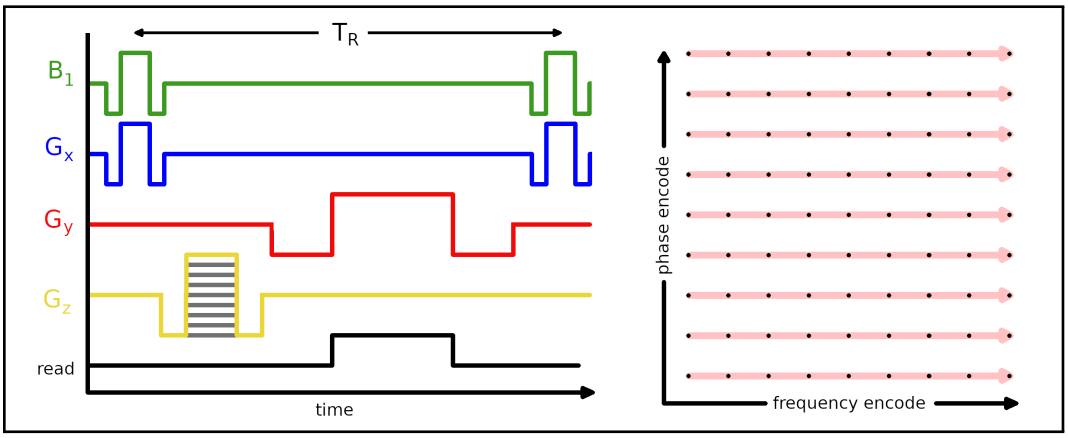

Figure 2.24: A simple illustration of a bSSFP pulse sequence, and the resulting path through k-space. Note that all of the net gradients are zero when evaluated over a single TR.

There are also several acceleration techniques that involve undersampling of k-space (see Figure 2.25). In partial Fourier reconstruction, a number of phase-encode rows away from the origin are left intentionally blank, and are either padded with zeroes or estimated based on assumptions about the symmetry of the image in k-space. In parallel imaging, multiple receive coils take images

over a coherently undersampled k-space, and the known spatial sensitivities of the coils are used to disambiguate the resulting aliasing. Compressed sensing is a technique that involves the undersampling of k-space in an incoherent manner, which overlays the image with diffuse noise resulting from incoherent aliasing. This noise can then be mitigated through a number of denoising strategies.

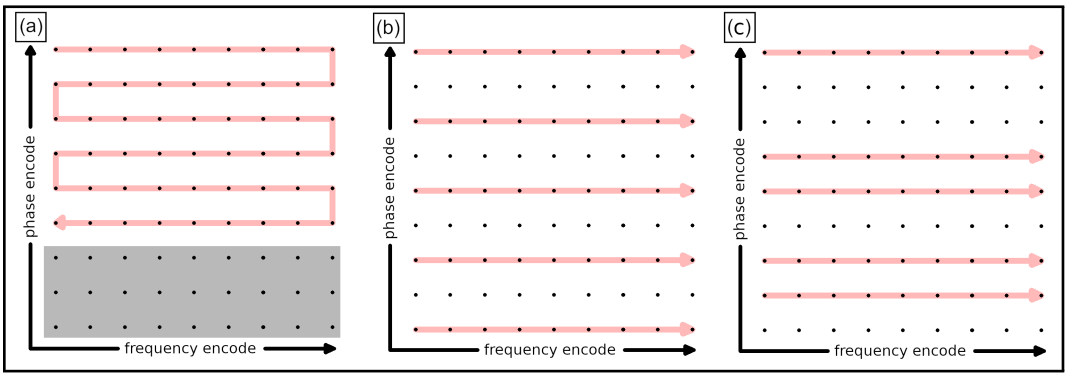

Figure 2.25: k-space undersampling strategies. (a): Partial Fourier reconstruction, wherein a continuous portion of k-space is left unmeasured and the missing points are either zero-padded or filled based on assumed symmetry. (b) Coherent k-space undersampling for parallel imaging. (c) Incoherent undersampling for compressed sensing.

2D real-time MRI combines a combination of fast pulse sequences and k-space undersampling to acquire 2D images over physiologically relevant time scales. For respiratory tracking specifically, a maximum tolerable system delay of 500 ms has been suggested[30]. Considering the other contributors to the system delay in linac-MR based tracking, this would require an MRI acquisition rate of 4 Hz or greater. Several linac-MR platforms have already exceeded this rate, and even faster acquisitions may still be attainable. For example, a combination of SSFP MRI and radial undersampling with parallel imaging has been shown to yield temporal resolutions as low as 20 ms[89] for 2D real-time MRI.

However, fast acquisition times are only one part of the equation. Many

imaging techniques based on k-space undersampling make use of computationally expensive iterative reconstruction algorithms. As a result, the gains made in acquisition time may be lost again (potentially several times over) during on-line reconstruction. For example, the 20 ms acquisition mentioned previously required 2.5 s of reconstruction per frame[89].

Pseudo 3D real-time MR tracking takes advantage of the fact that the slice select gradient can readily be applied at arbitrary angles, allowing for the alternation of imaging between orthogonal planes during 2D imaging to facilitate 3D tracking at a reduced framerate. True 3D real-time MR tracking is yet to be achieved. That said, an interesting approach called MR signature matching has recently been suggested, in which slow to acquire volumetric images are correlated to quickly obtainable motion signatures during an offline training phase, so that only the MR signatures need to be collected during treatment to estimate the image that would result[90].

For now, however, even for single-plane 2D imaging, the system delays resulting from linac-MR acquisition and reconstruction times result in a system delay that requires non-linear motion prediction. The next chapter describes in detail my contributions to improving the accuracy of these predictions and accelerating the speed at which they can be obtained.

Chapter 3

Accurate, On-Demand Neural Network Ensembles for Tumor Motion Prediction

A version of this Chapter has been submitted to the journal Medical Physics*, and is presently undergoing the peer-review process. Its current working title is "Accurate, On-Demand Neural Network Ensembles for Tumor Motion Prediction" by Neil W. Johnson, Keith Wachowicz, Satyapal Rathee, B. Gino Fallone and Jihyun Yun. It appears here in a modified form, both to match the formatting of this thesis and to reflect revisions suggested by the examining committee.*

Abstract

Purpose: To develop accurate, fraction-specific neural network-based tumor motion prediction for intra-fractional tumor tracking on hybrid linac-MR systems.

Methods: LSTM-RNNs are trained to predict the 3D position of abdominothoracic tumors based on their recent motion history. The effects of super-convergence regularization and ensemble methods on predictive accuracy and network training time are explored. Optimal hyperparameters for the LSTM-

RNNs are determined through a grid search of hyperparameter space. Inspired by extremely short training times, a novel strategy for network adaptation known as intermittent retraining (IR) is introduced and compared to online learning. Predictive accuracy is evaluated over 158 abdominothoracic tumor treatment fractions, each modified to simulate the range of image acquisition rates and system delays typical for hybrid linac-MR devices.

Results: Through implementing super-convergence regularization and selecting the computationally inexpensive optimized hyperparameters determined in this study, LSTM-RNN training times are reduced to 5 s per network on average. Ensembles of LSTM-RNNs improve prediction accuracy over individual networks at no cost to training time, provided adequate computing resources are available to perform training in parallel. IR adaptation outperforms online learning when super-convergence is implemented. A mean root mean square error of 0.35 mm – 0.79 mm (SD 0.26 mm – 0.49 mm) is achieved for prediction times ranging from 120 ms – 520 ms.

Conclusions: Predictive accuracy is improved relative to a comparable prior study that does not incorporate super-convergence regularization or ensemble methods and uses online learning, while training times are decreased by several orders of magnitude. To our knowledge, this marks the first time that accurate, on-demand, fraction-specific neural network-based tumor motion prediction has been made feasible.

3.1 Introduction

In DTTRT, there exists an inevitable latency between the imaging of a target and the adaptation of the therapeutic beam to its newly observed position.

This *system delay* consists of the sum of the time required to acquire and process each image, extract the shape and position of the tumor, then determine and perform the appropriate compensatory motion of the linear accelerator [42], [43], DMLCs[37]–[41], or patient support system[44]. Its magnitude depends on the imaging modality being employed as well as the relative maximum velocities of the tumor and the beam-steering hardware.

NifteRT is a novel implementation of DTTRT specific to hybrid linac-MR systems equipped with DMLCs[50], [51]. Under this approach, the excellent soft-tissue contrast provided by MR imaging allows for automatic image segmentation to rapidly extract the tumor position without the need for implanted markers, and the concurrently applied therapeutic beam can then be adapted on-the-fly to account for any detected motion. On the Alberta Linac-MR at the Cross Cancer Institute[26], [29], the system delay for performing nifteRT is anticipated to fall between 275 and 340 ms for a typical lung tumor[51], though system delays ranging from 120 ms to 520 ms have also been previously explored [66].

Since abdominothoracic tumors exhibit continual, often large-amplitude motion as a result of respiration[30], neglecting the system delay when treating them with nifteRT would result in a commensurate spatial lag of the therapeutic beam along their direction of motion. Precision treatment of these tumors therefore requires a method for accurately predicting respiration-induced tumor motion one system delay in the future, so that the compensatory hardware motion can be initiated at the appropriate time.

A variety of methods have been investigated for tumor motion prediction, including linear[54], [91], [92], sinusoidal [55] and polynomial regression [92], wavelet decomposition [92], [93], Kalman filters [53], [54], [67], [70], [94], [95], support/relevance vector machines [69]–[71], [96]–[98] and neural networks (NNs) [56], [63], [66], [72], [99]–[101]. Direct comparison of these ap-

proaches is complicated, since the results obtained in any individual study are inextricably tied to (1) the quality and motion characteristics the tumor motion dataset that was used; (2) the magnitude of the system delay being examined; and (3) the specific metric chosen for reporting accuracy. That said, a few studies have simultaneously evaluated multiple prediction techniques on a single motion dataset with consistent endpoints [91]–[93], [102]. Broadly speaking, these comparisons have favoured non-linear, adaptive methods to account for the non-linear and continuously evolving nature of free-breathing respiration. Specifically, adaptive NNs have consistently rated amongst the strongest performers, especially for system delays greater than 200 ms and patients with irregular respiratory patterns [102]. Of these NN-based approaches, LSTM-RNNs[50], [99], [103] lend themselves particularly well to time series prediction applications since they process their inputs serially, using memory gates to comprehend their temporal relationships.

There are several challenges inherent to applying NNs to tumor motion prediction: (1) the training process is typically time-consuming, which can be impractical clinically and may lead to performance degradation as the patient's respiratory patterns naturally evolve prior to treatment; (2) the accuracy is known to be heavily dependent on the chosen NN hyperparameters[64], which can be computationally expensive to optimize; (3) there is a relative paucity of training data available compared to traditional NN applications (especially for fraction-specific NNs), which can lead to network overfitting; (4) NNs are known to exhibit a strong dependence on their initial weights, especially when the number of training examples is limited[104].

There exist many common regularization methods that can be employed to mitigate both overfitting and instability, including dropout networks and lasso and ridge (also called L1 and L2) regularization. However, these regularization methods can substantially slow the training process. Recently, it has

been shown that a training strategy called super-convergence (a combination of aggressive learning rates and a simultaneous reduction in the number of training epochs performed) not only reduces training time, but also can act as a regularization method in and of itself, especially when limited training data are available[84]. It has also been shown that taking the consensus prediction of an ensemble of independently trained networks helps to mitigate both instability and errors arising from overfitting[104], [105].

This study introduces a novel approach that combines super-convergence regularization (using intelligent early stopping) with LSTM-RNN ensembles. We then describe a process for optimizing the hyperparameters of the networks that make up these ensembles, both on a fraction-specific and global basis. We introduce IR adaptation and compare the resulting predictive accuracy to that of a conventional adaptation strategy called online learning. Finally, an optimized version of our approach is evaluated across a range of acquisition rates and system delays expected to be relevant to nifteRT.

3.2 Materials and Methods

3.2.1 Abdominothoracic Tumor Motion Dataset

As in previous work from this research group [50], [66], the present study uses the tumor motion dataset of Suh *et al.*[106] 3D tumor trajectories from 46 patients with 50 tumors (33 lung, 17 retroperitoneal) over the course of 158 total treatment fractions were measured with the CyberKnife Synchrony Respiratory Tracking System (Accuray Incorporated, Sunnyvale, CA). This system forms and intermittently updates a correlation model between external surrogates and internal fiducials to estimate the tumor centroid position at the time points of the external surrogate signal. Tumor coordinates were reported at 40 ms intervals throughout all treatment fractions, which themselves ranged

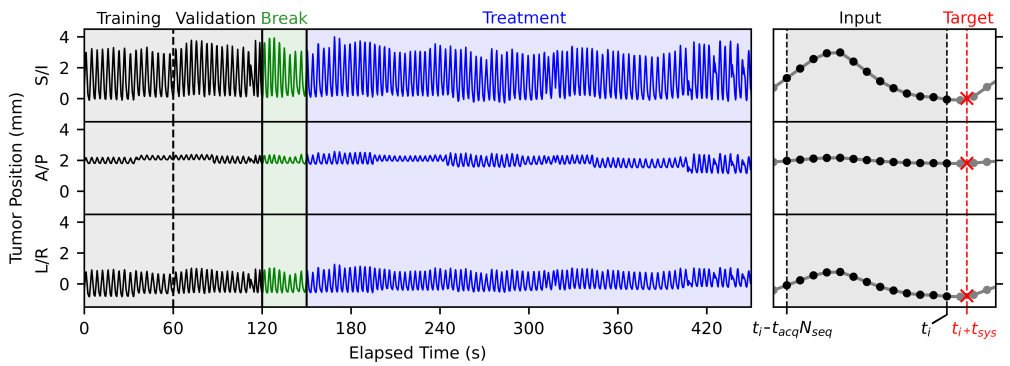

Figure 3.1: (Left panel) A sample 3D lung tumor trajectory (patient 14, fraction 51 in Table 3.2) with dataset divisions indicated. (Right panel) An example of an input/target pair with a system delay of 1.5× the acquisition time.

from 8 to 106 minutes in total length.

To simulate nifteRT treatment on a hybrid linac-MR system where the acquisition time (t_{acq}) is anticipated to fall between 120 ms and 280 ms [107], [108], the Synchrony data are resampled from every third to every seventh data point in order to generate multiple tumor trajectories representing t_{acq} values that span the range of interest in 40 ms steps (i.e., every third data point would give a simulated t_{acq} of 3×40 ms $= 120$ ms, while every seventh would give a simulated t_{acq} of 7×40 ms $= 280$ ms). To facilitate analysis, fractions are cropped to contain only the first 8 minutes of motion data, corresponding to the shortest measured fraction from the Suh *et al.* data. A sample 3D tumor trajectory is shown in the left panel of Figure 3.1, while Table 3.2 contains the tumor site and motion characteristics for all treatment fractions considered in this study. Generally, the dataset has an average mean motion amplitude of 4.96 mm (SD 3.1 mm, range 0.33 mm – 14.52 mm) and an average mean respiratory period of 3.88 s (SD 0.87 s, range 1.86 s – 7.06 s).

3.2.2 Mathematical Formulation

If the i^{th} 3D coordinate vector of the tumor centroid is denoted as $\vec{x}(t_i)$, a function f is desired that takes in its previously observed N_{seq} coordinates and predicts the tumor centroid position one system delay (t_{sys}) in the future, $\vec{x}_{pred}(t_i + t_{sys})$. That is,

$$f\left(\vec{x}\left(t_i - N_{seq}t_{acq}\right), \ldots, \vec{x}\left(t_i\right)\right) = \vec{x}_{pred}\left(t_i + t_{sys}\right) \tag{3.1}$$

Each tumor trajectory is therefore used to produce two sets of values: serial input sequences of length N_{seq} positions $[\vec{x}(t_i - N_{seq}t_{acq}), \ldots, \vec{x}(t_i)]$, and corresponding target coordinates one system delay after each input sequence ends, $\vec{x}_{true}(t_i + t_{sys})$. If t_{sys} is not an integer multiple of t_{acq}, linear interpolation is used to assign a value to $\vec{x}_{true}(t_i + t_{sys})$. For the analyses in Sections 3.3.1.1 to 3.3.2.1, it is assumed that $t_{acq} = 280$ ms and $t_{sys} = 320$ ms, conservative estimates of the capabilities of the Alberta linac-MR. A sample input/target pair are shown in the right panel of Figure 3.1.

3.2.3 Cost Function Selection

A method is required to evaluate the agreement between $\vec{x}_{pred}(t_i + t_{sys})$ and $\vec{x}_{true}(t_i + t_{sys})$, both as a cost function C to be minimized during NN training and as a metric for evaluating predictive accuracy. Root mean squared error (RMSE) is chosen for this study as it preferentially penalizes large misses, which cause a disproportionate increase in the standard deviation of the error (leading to increased safety margins in the formulations of ICRU 62[10], Van Herk[109] and Stroom[110]). Such large misses are also more likely to cause the therapeutic beam to temporarily intersect with a distant organ at risk, have a greater probability of tripping an error-catching algorithm designed to stop treatment if the predictive accuracy becomes unacceptable, and can be

indicative of overfitting.

In our formulation, the RMSE is mathematically defined as:

$$C = \frac{1}{j^{1/2}} \sqrt{\sum_{j} \left(\vec{x}_{pred,j} \left(t_i + t_{sys} \right) - \vec{x}_{true,j} \left(t_i + t_{sys} \right) \right)^2} \tag{3.2}$$

where j iterates over all the input/target pairs in the motion trace being examined.

In terms of reporting predictive accuracy, it is useful to define an amplitude-normalized cost function $\widetilde{C}$ to allow for a fairer comparison of performance between datasets with different motion characteristics. We first calculate the mean amplitude of respiratory motion for each 480 s treatment fraction ($\bar{A}$), and define the amplitude-normalized cost function $\widetilde{C}$ for that fraction as:

$$\widetilde{C} = \frac{1}{\bar{A} j^{1/2}} \sqrt{\sum_{j} \left(\vec{x}_{pred,j} \left(t_i + t_{sys} \right) - \vec{x}_{true,j} \left(t_i + t_{sys} \right) \right)^2} \tag{3.3}$$

3.2.4 Motion Dataset Division

In this study, each treatment fraction is divided into four components: (1) a training set of length t_{train} over which the cost function C_{train} is minimized through error backpropagation; (2) a validation set of length t_{val} over which the cost function C_{val} is continually evaluated during training in order to facilitate early stopping; (3) a break period of length t_{break} to simulate the time required for NN training and (4) a simulated treatment session of length t_{treat} over which the predictive accuracy C_{treat} (or $\widetilde{C}_{treat}$) is calculated. Example dataset divisions with $t_{break} = 30$ s are depicted in the left panel of Figure 3.1. In all experiments in this study, t_{treat} is taken as the final 300 s of motion in the 480 s trajectory, t_{train} and t_{val} are each 60 s in length, and t_{break} is varied between 5 s to 30 s, representing a range of achievable training times using the presented approach. For online learning, t_{break} is taken to be 0 s for the adaptation

process, since performing a single training epoch is nearly instantaneous.

To reduce computational expense, a sample of the total dataset consisting of the first treatment fraction for each of the first 10 patients is used for the optimization experiments in Section 3.3.1, while the complete patient dataset (including the optimization portion) is used for the subsequent experiments in Section 3.3.2. The optimization dataset has similar characteristics to the complete dataset in terms of the ratio of lung to retroperitoneal tumor sites (7:3 in optimization vs. 33:17 in complete), average motion amplitude (4.7 mm vs. 5.0 mm) and average respiration period (3.7 s vs. 3.9 s).

3.2.5 Neural Networks

The NNs used in this study have the following architecture, also illustrated in Figure 3.2: the first layer contains N_{seq} input neurons, each representing one previously measured 3D tumor position. These inputs feed sequentially into a stack of N_{HL} fully connected LSTM-RNN hidden computing layers, each having an identical activation function ϕ and identical width w_{HL} (with the addition of a single bias node). Where one LSTM-RNN layer feeds into another, the first layer passes a sequence of N_{seq} outputs to the next rather than a singular output, and the two are fully connected. The final computing layer is then densely connected to an output layer consisting of a single neuron representing the predicted 3D tumor coordinates. Since the output is unbounded and continuous, this neuron is equipped with a linear activation function.

LSTM-RNN training is accomplished through supervised learning, with the gradient of C_{train} with respect to network weights being calculated through backpropagation and optimized through gradient descent. We use the *Adam* algorithm[111] to manage the learning rate (LR) during training to ensure rapid convergence, with the aggressiveness of the learning process governed by the initial value of the LR, α_i. This optimizer has been shown to improve

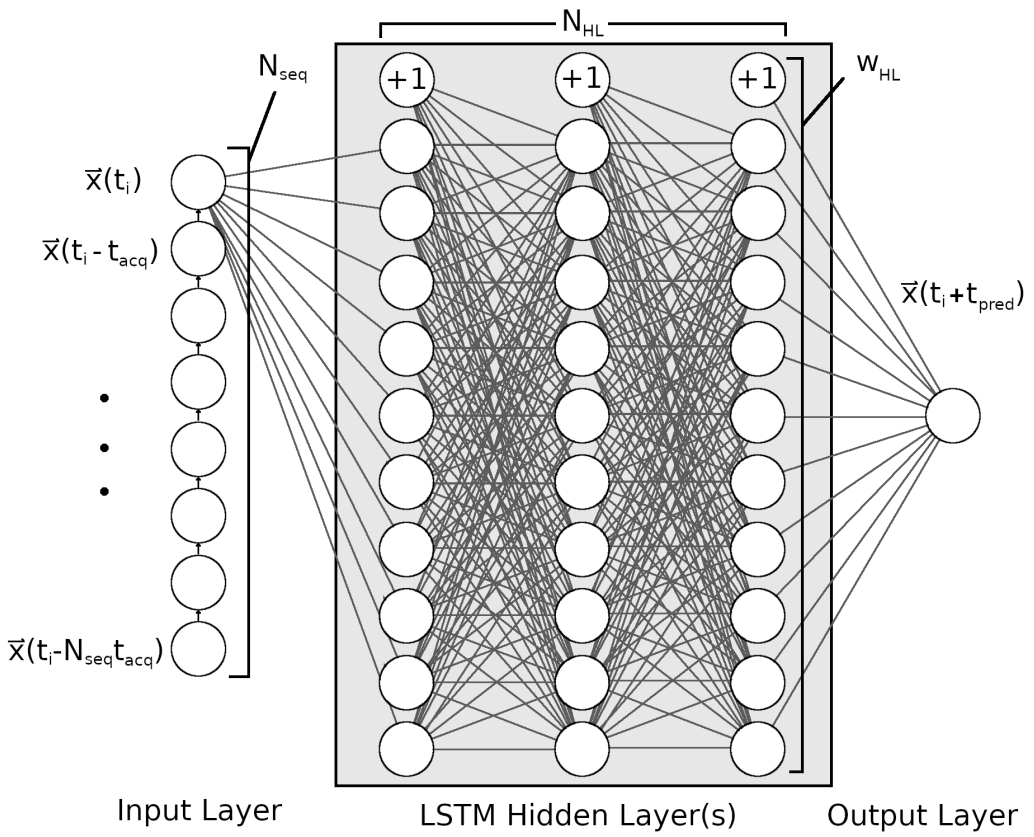

Figure 3.2: The architecture of the LSTM-RNNs used in this study, and the definition of each of the three architectural hyperparameters.

upon the standard gradient descent process for noisy cost functions resulting from sparse training data owing to its considerations of the first and second moments of the gradient.

3.2.6 Hyperparameter Optimization

Extracting the best possible performance out of a NN requires simultaneously optimizing a tuple of hyperparameters that determine the architectural and functional characteristics of the network and govern its training. In this study, a grid search optimization strategy is implemented, wherein each hyperparameter is assigned a range of possible values, and $\widetilde{C}_{treat}$ is exhaustively evaluated for each fraction in the optimization set over the entirety of hyperparameter space.

Four potential values of N_{seq} are evaluated per fraction: one half the measured average respiratory period, one full average respiratory period, two average respiratory periods, and a constant time length of 3.6 s (approximately the average respiratory period across the full motion dataset, following a previous publication[9]). N_{HL} is limited to between 1 and 3, after having observed during initial experiments that both the accuracy and training time of the networks degrade with increased network depth. For a similar reason, w_{HL} is chosen as either 10, 20 or 50 neurons, having observed a decrease in performance and simultaneous increase in training time at larger widths. The activation function ϕ is chosen from the sigmoidal and tanh activation functions commonly used for LSTM-RNNs, as well as the ReLU and leaky ReLU (LReLU) activation functions, which are less computationally expensive and therefore usually result in faster training. Since super-convergence regularization is being implemented, the only hyperparameters related to training are the initial learning rate α_i and the number of training epochs performed. Candidate values for α_i are determined in Section 3.3.1.1 using the process outlined in Section 3.2.7, after which it can be optimized together with the rest of the hyperparameters. The number of training epochs performed is determined on a per-network basis by the parameters that govern the early-stopping process. These are described in Section 3.2.8, while Section 3.3.1.2 details their optimization.

3.2.7 Initial Learning Rate Determination

A process is required to identify values of α_i that are sufficiently aggressive to reap the benefits of super-convergence, but not so aggressive as to result in divergence. Previously, a method was introduced to determine appropriate upper and lower LR bounds for training NNs with cyclic LRs that has since been incorporated into a TensorFlow callback called LRFinder[112]. In the

present study, we adapt this process to determine a range of candidate values for α_i. Briefly, training of the network is performed with a LR that exponentially increases over several orders of magnitude. Generally, a plot of C_{train} vs. LR (see Figure 3.3) will exhibit a plateau region (where the LR is low enough that a single training epoch causes negligible improvement to C_{train}), followed by a decline (as the network parameters begin converging toward their optimal values), followed by a steep inflection (where the LR is too large, resulting in divergence). By selecting α_i on the decline and prior to the inflection point, the effects of super-convergence can be maximized while avoiding divergence.

There is likely to be some variation in this optimal value of α_i between different network hyperparameter configurations, different motion fractions and different random initializations of network weights. Therefore, for each combination of N_{HL}, w_{HL}, ϕ and N_{seq} in the hyperparameter grid and each fraction in the optimization set, LRFinder is used to generate 25 LR curves for networks initialized with different weights, for a total of 3.6×10^3 (3 values each for N_{HL} and w_{HL}, 4 each for ϕ and N_{seq}, 25 random initial weight configurations per fraction and 10 fractions). Any dependence of α_i on other hyperparameters can then be determined from these curves, and candidate values can be identified for hyperparameter optimization.

3.2.8 Early Stopping

Two mechanisms are used to enforce early stopping. First, C_{val} is evaluated after every training epoch and when it fails to improve for a set number of epochs (the *patience* of the early-stopping process), training is halted and the network is restored to the state that resulted in the minimum observed C_{val}. Here, it is assumed that a network with a minimized C_{val} should also have a minimized C_{treat}, since both depend on the network's ability to generalize to motion data outside the training set. Using too small a patience may stop the

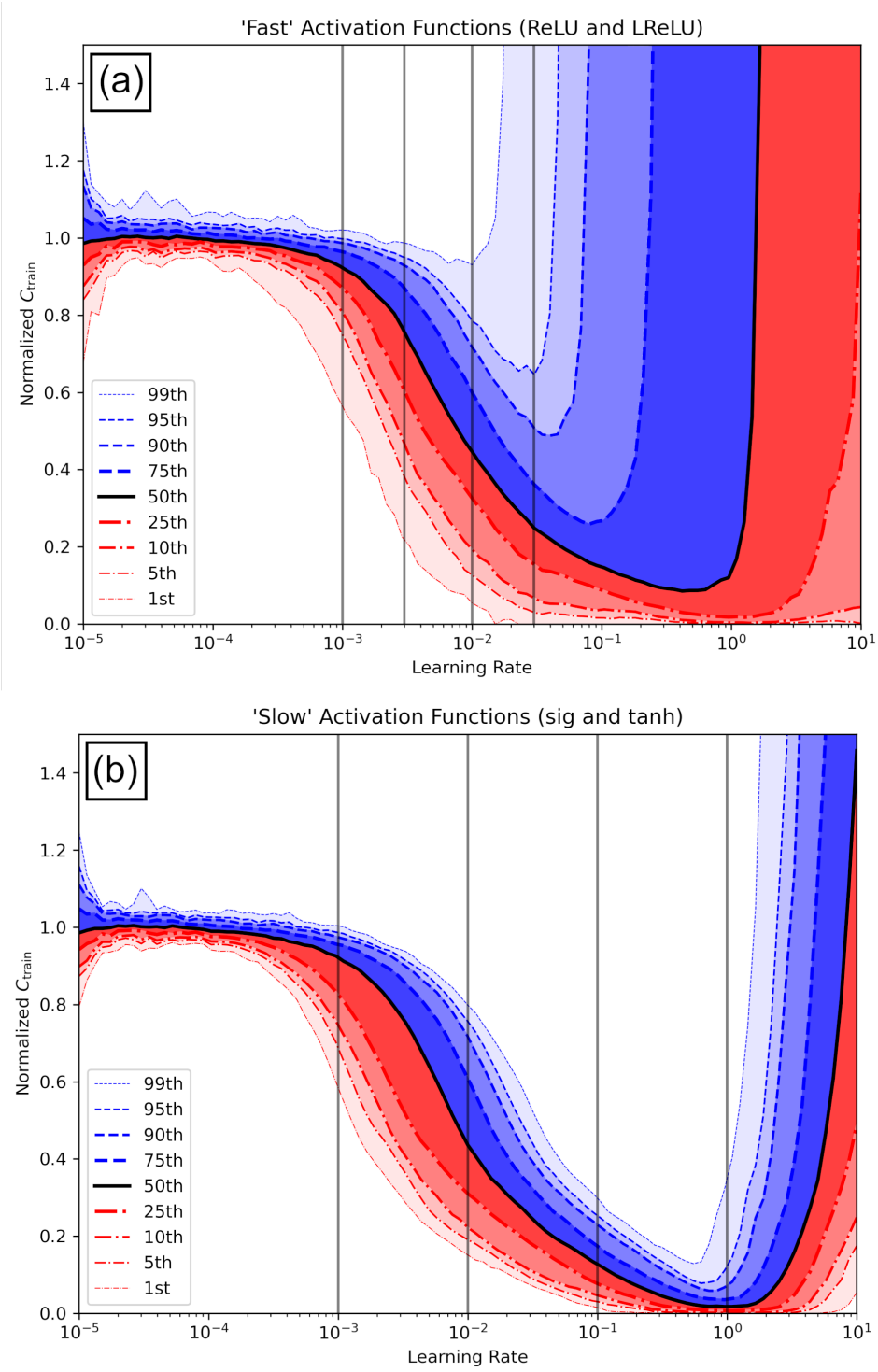

Figure 3.3: Percentile LR curves for 'fast' and 'slow' activation functions, each derived from 1.8×10^3 runs of the LRFinder algorithm (10 runs at each point in hyperparameter space). Selected α_i hyperparameter grid points are indicated with vertical lines.

training process before C_{val} is minimized, while using too large a patience may result in extended training times with limited or no improvement in C_{treat}, as well as an increased likelihood of reaching overfit solutions. Second, training is automatically halted after reaching a pre-determined maximum number of epochs e_{max}, even if C_{val} is still improving. This helps to prevent slowly converging networks from greatly extending the required training time. An optimal value for e_{max} would adequately allow the best-performing networks to reach their minimum C_{val}.

For every grid point in hyperparameter space and every fraction in the optimization set, 10 randomly initialized networks are trained for 300 epochs, which is about 10 times the number of epochs required for the fastest networks to converge in initial experiments. The minimum C_{val} achieved and the average number of epochs at which training ends are then simulated for different halting conditions (patience ranging from 5 to 100 epochs in 5 epoch increments and e_{max} ranging from 50 to 300 epochs in 50 epoch increments) so that the trade-off between predictive accuracy and training speed can be better understood.

3.2.9 Ensemble Construction

From Section 3.3.1.3 onward, network ensembles are generated by independently training N_{ens} networks, each starting from a different random initial weight configuration. Simply taking the mean of these N_{ens} individual network predictions is not likely to be beneficial, since there is still a chance that one or more networks may have either diverged or overfit during training, both of which could result in extreme predictions that would dominate the mean. Trimmed means, in which the most extreme predictions are rejected before the mean of the ensemble is calculated, have been suggested to work around this problem[105], but simply taking the median prediction (in each cardinal

dimension) of the N_{ens} networks should yield similar results. This approach should also be more robust than a trimmed mean against multiple networks making extreme errors in the same direction, especially for smaller ensembles. An illustration of the construction and application of ensembles in this study is provided in Figure 3.4.

3.2.10 Online Learning and Intermittent Retraining

Online learning, a common approach to network adaptation[113], involves performing a single backpropagation-based network weight update based on the error generated by each new acquired/predicted target pair as soon as it becomes available. The state of the optimizer is carried over from the initial training and between subsequent adaptations. This poses two issues: (1) when super-convergence regularization is employed, the LR may be left at an inappropriately large value at the end of initial training, leading to overcorrection during adaptation; (2) this approach may emphasize learning the initial training set over the newest, most relevant motion data, since the initial training set is reinforced over multiple training epochs.

In this study, the short training times that we achieve through super-convergence regularization allow us to propose a novel adaptation strategy called IR in which networks (or, in this case, ensembles) are fully retrained at intervals of t_{break} throughout the treatment, using the most recent motion data for training and validation (see Figure 3.5). In practice, the minimum t_{break} is dictated by the slowest network to converge in each ensemble, assuming training is done in parallel. Each time one training session ends, the newly-prepared ensemble is substituted in as the predictor for treatment, and another training session can begin in the background. The first ensemble is always randomly initialized at the outset of training, while subsequent networks can either be initialized randomly or continue training from the most recently

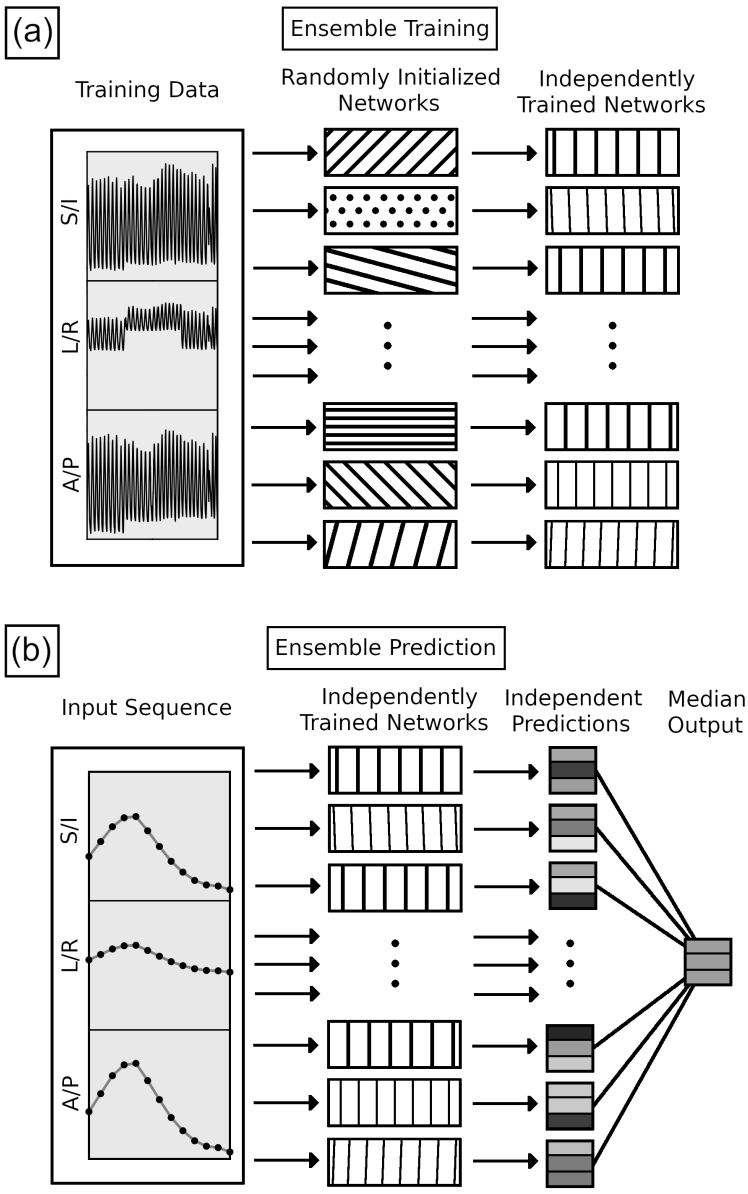

Figure 3.4: (a) N_{ens} randomly initialized networks are trained independently to create an ensemble of unique predictors. (b) The median output of the ensemble (in each of the three cardinal directions) is taken as the ensemble prediction for each input sequence in the treatment fraction.

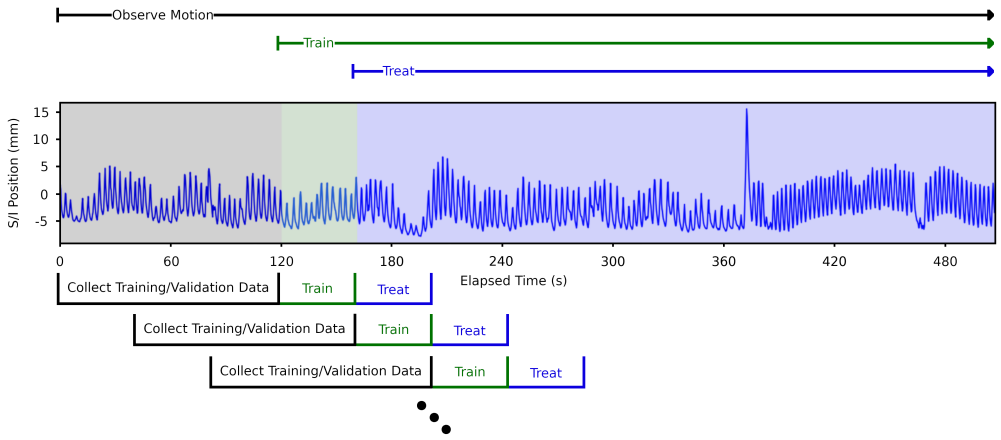

Figure 3.5: An illustration of the intermittent retraining process. During ensemble training, new data are collected that are more relevant to future motion than the training/validation data that inform the ensemble. Once treatment begins, a new ensemble is trained in the background on a training/validation set including these new data. The length of the training interval is the same as t_{break}. Here, only the S/I tumor motion is shown for illustrative purposes.

obtained solution. The latter approach may improve performance throughout treatment as the NN solution will be based on an ever-expanding training set, with an emphasis on the most recent and therefore relevant motion data.

3.2.11 Software and Hardware

The code for this project is written in Python 3.7[114], using the Keras API to interact with the TensorFlow 2.1.0 machine learning backend[115]. Learning rate optimization is performed using the LRFinder callback[112], while early stopping is implemented using the built-in EarlyStopping callback available with TensorFlow 2.1.0. So that the results in this study may be replicated, the pseudo-random number generator seed used for initializing network weights is always iterated sequentially from 0 (for example, in Section 3.3.1.3 where 50 networks are used to make network ensembles, the 50 seeds used are $[0, 1, \ldots, 49]$). Training is performed on an Intel Core i9-7900X CPU with a base speed of 3.30 GHz. Inter- and intra-operation parallelism are both set to 1, so each training instance is restricted to a single virtual processor.

3.3 Results

3.3.1 Optimization Subset Experiments

3.3.1.1 Learning Rate Determination

In initial experiments, a LR of 10^{-5} was found to be well into the plateau region of the learning curve for all tested fractions and hyperparameter combinations, and a LR of 10^{1} was observed to result in near universal divergence. These values were therefore selected as the respective minimum and maximum LR bounds. Since we are only interested in the transition point between the decline and divergence spike and not the absolute value of the training accuracy, each curve in Figure 3.3 is normalized such that the mean of the first 18 points in LR space (from 10^{-5} to 10^{-4}) is 1 and the minimum C_{train} over LR space is 0. These curves are generally observed to be equivalent across all fractions in the optimization set and across most hyperparameters, with the exception of the activation function. On average, the tanh and sigmoid activation functions result in LR curves with much longer sloped regions and later divergence spikes than ReLU and LReLU (see Figure 3.3), most likely as a result of their smaller gradients (for example, the maximum gradient of the sigmoid function is 0.25 compared to 1.0 for ReLU). Interestingly, these 'slower' activation functions also exhibit generally less spread in loss, which may indicate more uniform convergence and less dependence on initial weights. This characteristic is usually desirable when training individual networks, but a lower degree of network diversity may be disadvantageous for ensemble-based approaches since their predictions are less likely to be uncorrelated, limiting the statistical benefit of pooling multiple predictions.

In Figures 3.3(a) and 3.3(b), percentile plots of LR curves are shown for the 'fast' and 'slow' activation functions, respectively. Each plot is based on the results of 1.8×10^3 total LR curves (since both only contain results from

two of four activation functions). For each 'speed' of activation function, two key values of α_i are identified: (1) the LR at which all networks appear to begin converging (i.e., the 99^{th} percentile line leaves the plateau), and (2) the LR at which 10% of the networks have started diverging (i.e., the 90^{th} percentile line begins inflecting upward). The α_i grid points are then spread evenly over this range. For the 'fast' activation functions, this corresponds to $[1 \times 10^{-3}, 3 \times 10^{-3}, 1 \times 10^{-2}, 3 \times 10^{-2}]$, and for the 'slow' activation functions they are $[1 \times 10^{-3}, 1 \times 10^{-2}, 1 \times 10^{-1}, 1 \times 10^{0}]$, as indicated in Figures 3.3(a) and 3.3(b) with dashed lines.

3.3.1.2 Early Stopping Parameters

We do not desire to optimize the patience and e_{max} early stopping parameters to yield the minimum C_{val} across all hyperparameter space, since this could be disproportionately influenced by networks that converge too slowly to benefit from super-convergence. Instead, it is better to optimize the patience and e_{max} settings only for the hyperparameter configurations that yield the smallest minimum C_{val}. For the ten best performing hyperparameter settings in terms of C_{val}, it is found that a patience of 20 epochs and an e_{max} of 100 results in very fast convergence (training halts at the 70^{th} epoch on average, corresponding to about 5 s training time on our hardware) with an increase of only 8% to the mean minimum C_{val} observed compared to training every network for the full 300 epochs (which would take about four times longer, on average). We therefore use these halting conditions for subsequent experiments.

3.3.1.3 Optimal Network Hyperparameters

For hyperparameter optimization, ensembles of $N_{ens} = 50$ networks are trained for each grid point in hyperparameter space and the amplitude-normalized cost function $\widetilde{C}_{treat}$ is calculated from their median output. The training step

of this process is computationally expensive, requiring 2.88×10^4 networks (50 networks and 576 hyperparameter grid points) per fraction to be trained. Adaptation strategies are not implemented during hyperparameter optimization, so the evaluation of $\widetilde{C}_{treat}$ is restricted to the first 30 seconds of motion during the treatment fraction in order to optimize immediate performance rather than longer-term stability.

It can be anticipated from the analysis in Section 3.3.1.1 that there will be an interdependence between α_i and ϕ, so the effect of varying α_i is treated separately for the 'fast' and 'slow' activation functions (see Figure 3.6(a)). Interestingly, $\alpha_i = 1 \times 10^{-2}$ yields the best accuracy for both types of activation function (though $\alpha_i = 3 \times 10^{-3}$ is a close second for the 'fast' ones, but is less preferable because it results in longer training times). To facilitate analysis, only this best-performing α_i is considered during the optimization of the rest of the hyperparameters.

It is clear from Figure 3.6(b) that $N_{HL} = 1$ is strongly preferred, and that networks with the sigmoid activation function perform worse on average than the alternatives considered in this study. Mean performances are similar across all tested values of w_{HL} and N_{seq}, though there is an indication that $w_{HL} = 10$ may be less preferential in some circumstances given its more prominent tail in the direction of larger error.

Analyzing the best performing hyperparameter configuration on a fraction-specific basis over the optimization set (see Table 3.3), $N_{HL} = 1$, $w_{HL} = 50$ and $N_{seq} = 1$ respiratory cycle appear slightly more often than the alternatives, but almost every other value for each hyperparameter appears at least once in the list (with the exception of $N_{seq} = 3.6$ s). Additionally, there does not appear to be a discernible relationship between tumour site, motion amplitude nor respiratory rate and any of the individual hyperparameters. However, when averaged over all fractions in the optimization set (Table 3.4), 7 of the 10 best-

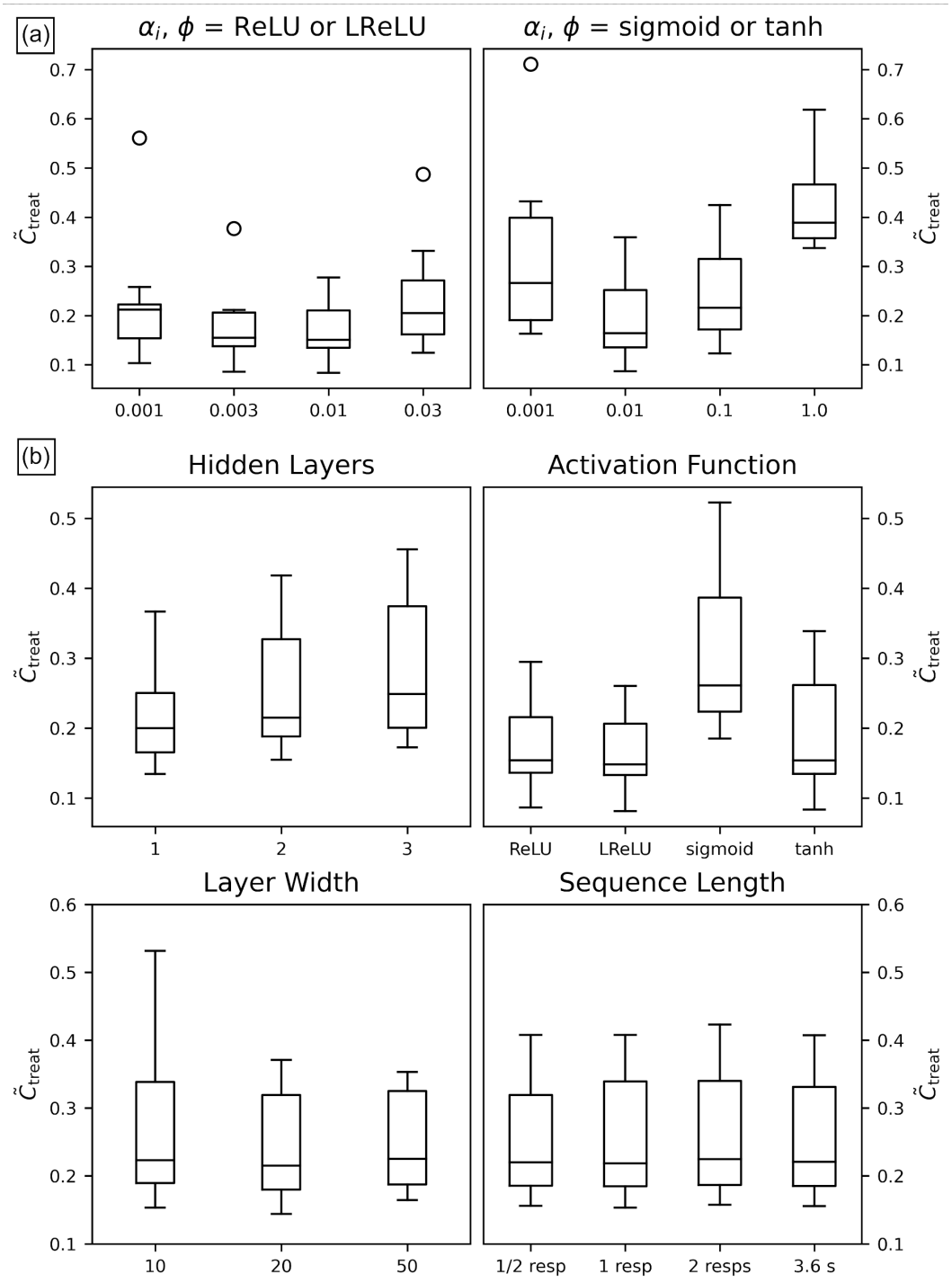

Figure 3.6: (a) Mean $\widetilde{C}_{treat}$ when varying α_i for 'fast' and 'slow' activation functions. (b) Mean $\widetilde{C}_{treat}$ when varying the rest of the free hyperparameters, with α_i fixed at its optimal values for each family of activation functions.

performing global hyperparameter configurations have ϕ = LReLU (the rest are ReLU), 8 of 10 have $N_{HL} = 1$, all have $w_{HL} = 20$ or 50, and 6 of 10 have $N_{seq} = 3.6$ s. Taking a global optimal hyperparameter configuration of ϕ = LReLU, $N_{HL} = 1$, $w_{HL} = 20$, $\alpha_i = 1 \times 10^{-2}$ and $N_{seq} = 3.6$ s is determined to result in only a 10% average decrease in accuracy relative to optimizing hyperparameters on a fraction-specific basis, does not require measuring the average respiratory rate prior to treatment (as using N_{seq} = one half, one or two respiratory periods would), and comes with a 2.5× acceleration to training relative to the average over all hyperparameter configurations (4.5 s versus 12.0 s). This is an important result, as it indicates that global LSTM-RNN hyperparameters perform comparably to patient-specific ones under our approach, obviating the need for the computationally expensive hyperparameter optimization step in the future.

3.3.1.4 Effect of Ensemble Size

Assuming the individual networks composing the ensemble are independent, increasing N_{ens} will simultaneously improve the predictive accuracy and reduce its variance as a result of better statistics. Practically, increasing N_{ens} demands either a larger number of processors dedicated to training, or possibly extending training times to allow for one processor to serially train several networks.

In order to understand the effect of ensemble size on network performance, 100 networks are trained on the optimization patient set using the global optimal hyperparameters. Ensembles varying in size from $N_{ens} = 1$ to $N_{ens} = 75$ are then randomly selected from the pool of networks 100 times, so that the mean and variance of the resulting $\widetilde{C}_{treat}$ values can be determined at each ensemble size.

Another common approach to mitigating initial weight dependencies is to

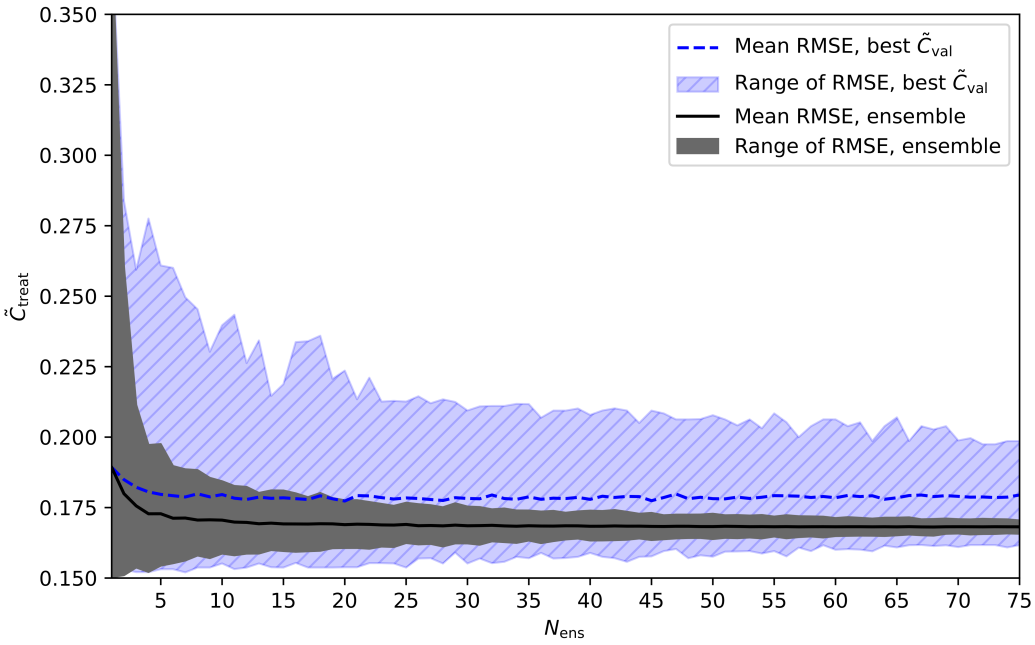

Figure 3.7: Effect of varying N_{ens} on mean and range of $\widetilde{C}_{treat}$ based on randomly selecting trained networks from a pool of 100, using 100 different configurations per ensemble size. Also shown are the mean and range of $\widetilde{C}_{treat}$ if only the network with the best C_{val} in each ensemble is used for prediction.

train multiple networks, then only use the network that yielded the minimum C_{val} for motion prediction[66]. This approach results in a negligible reduction in required computational resources compared to an ensemble consisting of the same number of networks, since the training step is by far the most computationally expensive. For each ensemble generated in Figure 3.7, the accuracy of the individual network that had the best performance over the validation set is also recorded for comparison.

On average, ensembles yield about 5% better $\widetilde{C}_{treat}$ than selecting the network with the best C_{val}, given the same number of random initializations. The range of $\widetilde{C}_{treat}$ observed is also considerably smaller, implying more reliable accuracy. By about $N_{ens} = 25$, the performance of the ensembles has largely saturated and the range of $\widetilde{C}_{treat}$ falls below 10% of its mean. Increasing N_{ens} to 50 and 75 reduces this range to 5% and 2.5% mean $\widetilde{C}_{treat}$, respectively.

This result indicates that the largest possible N_{ens} given the hardware available should always be used to obtain more robust results, but to limit computational expense in this study N_{ens} is set to 25 for the experiments in Section 3.3.2.

3.3.2 Full Motion Dataset Experiments

3.3.2.1 Adaptation Strategies

Using the optimal global hyperparameter configuration determined in Section 3.3.1.3, ten different adaptation strategies are evaluated over the entire motion dataset: no adaptation (as a control), online learning, and IR at 5 s, 10 s, 20 s or 30 s retrain intervals with network weights initialized either from scratch or from the previous solution prior to retraining. The results of this experiment are shown in Figure 3.8(a). On average, online learning results in an 11% improvement to $\widetilde{C}_{treat}$ over a non-adaptive approach. In all cases IR performs better, yielding between 16% and 25% improvement over a non-adaptive approach depending on the retraining interval. Initializing each network using the most recent available solution reduces the achievable $\widetilde{C}_{treat}$ by about 5% compared to randomly initializing the networks from scratch each time, with the additional benefit of a further improvement to average training times during adaptation (1.6 s per network versus 5.2 s).

In Figure 3.8(b), $\widetilde{C}_{treat}$ is displayed as a function of elapsed treatment time for four of the tested adaptation strategies (only IR at 10 s retrain intervals is displayed). As can be expected, not using any adaptation strategy results in a considerable loss in predictive accuracy over the course of the treatment (nearly 40% over 300 s). Online learning yields a worse mean $\widetilde{C}_{treat}$ over the first 90 s of treatment than no adaptation strategy, which as mentioned earlier is likely due to too large a LR being carried over into adaptation from the initial training run. By the end of the 300 s treatment, online learning and IR

from scratch perform equally well. IR from the previous solution consistently results in the best $\widetilde{C}_{treat}$ at all time points, with its performance appearing to improve over the course of the 300 s treatment. This may be a real effect from the networks accruing more training data as the treatment progresses, but it could also be an artefact of inconsistencies in the average motion regularity. Regardless, these results indicate that, if feasible based on training times, IR from the previous solution should always be implemented as an adaptation strategy instead of online learning, at least when super-convergence is being employed.

3.3.2.2 Effect of Acquisition Time and System Delay

Table 3.5 reports the mean $\widetilde{C}_{treat}$ taken over the complete motion dataset when varying t_{acq} and t_{sys} over their range of anticipated values, using 25-member ensembles with globally optimal hyperparameters and IR with a conservative 30 s retrain interval. $t_{acq} = 200$ ms yields the lowest mean $\widetilde{C}_{treat}$ when evaluated across the full range of t_{sys}, but overall there is little observable dependence on t_{acq} except for slight ($< 5\%$) increases at 120 ms and 280 ms. A slightly larger mean $\widetilde{C}_{treat}$ at a lower t_{acq} seems at first glance counterintuitive, but it is most likely because N_{seq} increases with sampling rate, making it easier for the network to find spurious connections between inputs and outputs during training. Conversely, at higher t_{acq} values, the number of examples in the training set decreases, which can lead to an increased propensity for overfitting. In Table 3.1, the mean $\widetilde{C}_{treat}$ across the complete motion dataset and the full range of t_{acq} is reported as a function of t_{sys}. As expected, the predictive accuracy falls off with increased system delay, roughly doubling when t_{sys} is tripled over the range of interest. Efforts should therefore be focused on increasing MR frame rates not because the more frequent imaging itself leads to better predictions, but because it will reduce the portion of t_{sys} stemming from acquisition time.

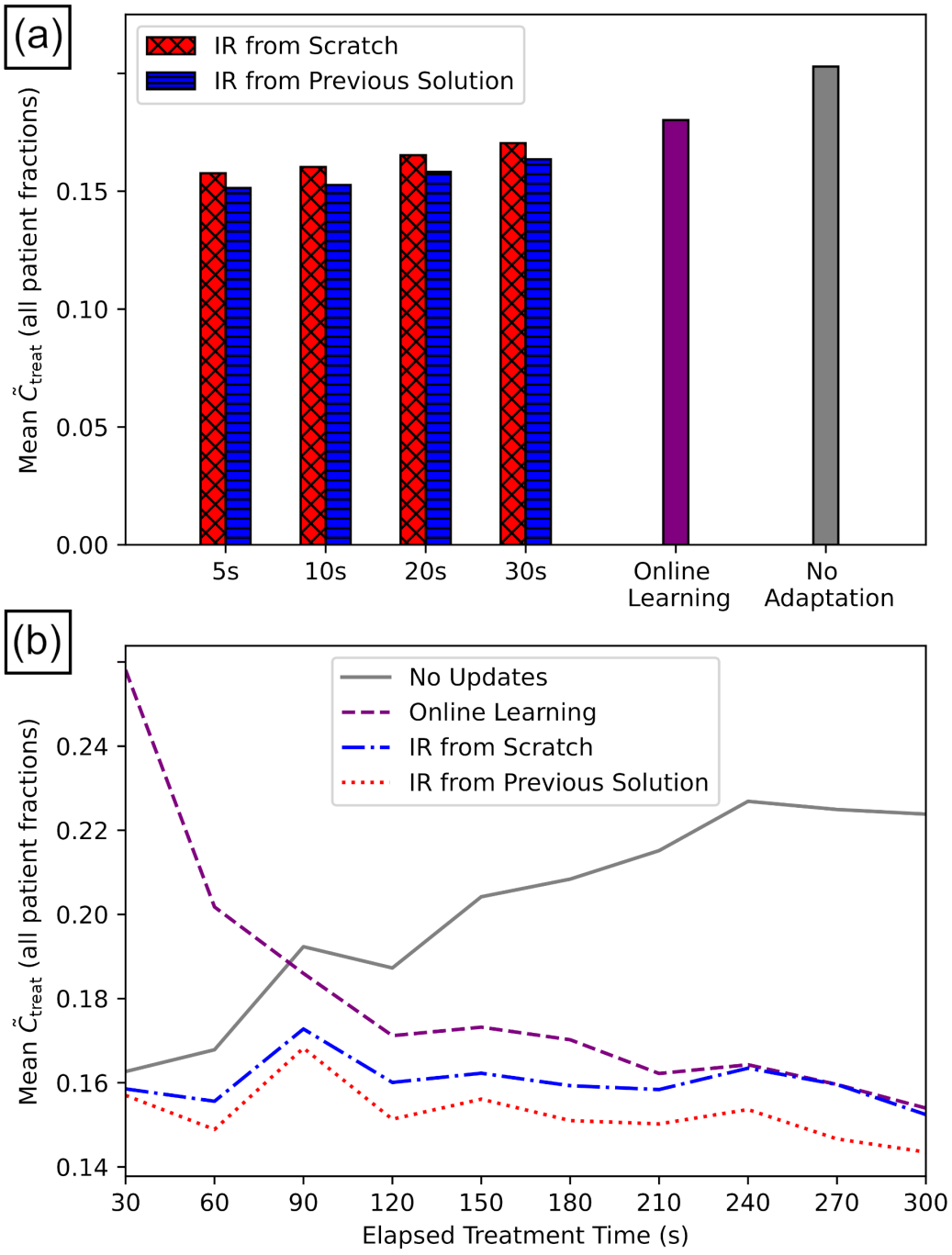

Figure 3.8: (a) Mean $\widetilde{C}_{treat}$ for several adaptation strategies: no adaptation (as a control), traditional online learning, and the adaptation strategy proposed in this study at various retraining intervals, with the adapted networks either starting from a random initialization (from scratch) or from their previous solution. (b) A plot of predictive accuracy as a function of treatment time, showing the decay in accuracy with no adaptation strategy present, the effects of the problematic inherited learning rate for online learning adaptation, and the superior performance of IR when the previous solution is used for network initialization. The IR retraining interval in this case is 10 s.

$\mathbf{t_{sys}}$ **(ms)**	**120**	**160**	**200**	**240**	**280**	**320**	**360**	**400**	**440**	**480**	**520**
C_{treat} (mm)	0.35	0.38	0.45	0.49	0.55	0.58	0.63	0.67	0.71	0.76	0.79
Std. Dev. C_{treat} (mm)	0.26	0.25	0.29	0.31	0.35	0.36	0.38	0.41	0.43	0.46	0.49
$\tilde{C}_{treat}$	0.09	0.09	0.11	0.12	0.13	0.14	0.15	0.15	0.16	0.17	0.18
Std. Dev. $\tilde{C}_{treat}$	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.08

Table 3.1: Mean and standard deviation of C_{treat} and $\widetilde{C}_{treat}$ across all treatment fractions and all values of t_{acq} as a function of t_{sys}.

3.4 Discussion

Compared to previous work using the same motion dataset and LSTM-RNN networks[50], we report 30% reductions in both mean C_{treat} and the standard deviation of C_{treat} through our implementation of super-convergence regularization, ensemble methods, hyperparameter optimization and IR adaptation. Importantly in terms of the practicality of NN-based prediction, we also report an associated acceleration of the training process of several orders of magnitude, from a few hours per network[50] to about 5 s (and $<$ 2 s for adaptation). We find comparable performance between global hyperparameters and fraction-specific ones, obviating the need for the computationally expensive hyperparameter optimization step.

To our knowledge, this makes accurate on-demand, fraction-specific NN-based respiratory motion prediction feasible for the first time, which will be highly beneficial to DTTRT devices like linac-MRs that have inherently long system delays.

3.5 Conclusion

The accuracy that we report assumes continual recording of 3D tumor coordinates throughout treatment. However, most linac-MRs are not currently able to acquire 3D cine-MR at the rates presented in this work. Our approach could be easily adapted to 2D motion for beam's eye view imaging since motion

in each dimension is treated independently. However, doing so may require acquiring a new training and validation set every time the treatment angle is changed, since the 2D motion from one plane could not be used to fully inform the 2D motion in another. That said, several authors are working on accelerated imaging techniques for linac-MRs[116], so 4 fps 3D cine-MR could be on the near horizon.

One open question regarding tumor motion prediction is how to assign appropriate treatment margins prior to treatment, given that the predictive accuracy itself can be difficult to predict[117]. We observe a large standard deviation of $\widetilde{C}_{treat}$ relative to its mean, which implies that setting margins based on tumor motion amplitude alone is not satisfactory, probably because it fails to account for the patient's respiratory regularity. It might, however, be more feasible to associate validation error during a training session to the performance in the subsequent treatment segment. This would allow for the automatic setting of margins as the amplitude and regularity of a patient's respiration changes, and also for the halting of treatment if the safety margins grow too large.

Acknowledgements

The authors gratefully acknowledge Dr. Paul Keall, Dr. Yelin Suh, and Dr. Sonja Dieterich from Stanford University for the tumor motion data used for this study. We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) under grant RGPIN-2016-05185 as well as the Canadian Institutes of Health Research (CIHR) under grant 437221.

Conflicts of Interest

The authors have no relevant conflicts of interest to disclose.

Supplemental Data

The following Tables were included as supplemental data for the submitted manuscript.

					t_{acq} = 280 ms t_{sys} = 320 ms	
Patient	Fraction	Tumor Location	Mean Amplitude (mm)	Mean Period (s)	C_{train} (mm)	$\tilde{C}_{train}$
1	**1**	**Lung Hilum Right**	**6.95**	**2.63**	**1.10**	**0.159**
1	2	Lung Hilum Right	7.01	2.72	0.71	0.102
1	3	Lung Hilum Right	6.75	2.93	0.80	0.119
2	**4**	**Pancreas**	**7.74**	**3.99**	**0.59**	**0.076**
2	5	Pancreas	5.04	3.48	0.40	0.080
2	6	Pancreas	6.76	3.37	0.37	0.054
3	**7**	**Lung RML**	**3.11**	**2.99**	**0.46**	**0.148**
3	8	Lung RML	3.38	3.07	0.86	0.266
3	9	Lung RML	2.67	2.94	0.38	0.146
3	10	Lung RML	4.11	2.62	0.51	0.124
4	**11**	**Pancreas**	**2.83**	**3.92**	**0.32**	**0.113**
4	12	Pancreas	6.49	7.06	0.43	0.067
4	13	Pancreas	2.43	4.02	0.25	0.107
4	14	Pancreas	4.53	5.29	0.33	0.073
4	15	Pancreas	5.30	4.22	0.35	0.066
4	16	Pancreas	3.18	4.08	0.26	0.081
5	**17**	**Lung LUL**	**4.82**	**4.14**	**0.36**	**0.075**
5	18	Lung LUL	3.01	3.95	0.37	0.128
5	19	Lung LUL	2.52	3.84	0.42	0.171
5	20	Lung RLL	8.61	3.83	0.67	0.078
5	21	Lung RLL	9.39	3.55	0.78	0.084
5	22	Lung RLL	8.22	3.24	1.27	0.155
5	23	Lung Bronchus Right	4.12	4.26	0.34	0.082
6	**24**	**Lung LUL**	**6.72**	**3.49**	**0.70**	**0.105**
6	25	Lung LUL	6.67	3.57	0.73	0.109
7	**26**	**Retroperitoneum**	**2.29**	**4.34**	**0.33**	**0.147**
7	27	Retroperitoneum	1.22	4.68	0.12	0.096
7	28	Retroperitoneum	1.19	4.41	0.18	0.151
7	29	Retroperitoneum	1.63	3.97	0.14	0.086
7	30	Retroperitoneum	3.50	4.46	0.41	0.119
8	**31**	**Lung LUL**	**2.69**	**4.10**	**0.36**	**0.137**
8	32	Lung LUL	1.63	4.58	0.33	0.212
8	33	Lung LUL	1.60	3.91	0.27	0.172
9	**34**	**Lung RUL**	**4.28**	**4.08**	**0.46**	**0.108**
9	35	Lung RUL	4.57	4.38	0.55	0.121
9	36	Lung RUL	3.67	4.46	0.80	0.220
10	**37**	**Lung LUL**	**3.95**	**3.01**	**0.49**	**0.126**
10	38	Lung LUL	3.91	2.91	0.83	0.225
10	39	Lung LUL	7.71	3.13	1.07	0.140
11	40	Lung RML	4.86	5.13	0.59	0.121
11	41	Lung RML	9.42	4.84	0.57	0.061
11	42	Lung RML	7.88	4.56	0.54	0.069
12	43	Lung RML	2.18	4.48	0.38	0.177
12	44	Lung RML	2.95	5.03	0.32	0.111
12	45	Lung RML	1.96	4.91	0.30	0.156
13	46	Lung RML	3.51	3.24	0.61	0.180
13	47	Lung RML	3.79	3.29	0.30	0.078
13	48	Lung RML	4.90	3.48	0.39	0.079
14	49	Lung RUL	3.25	3.18	0.33	0.101
14	50	Lung RUL	3.26	3.82	0.30	0.094
14	51	Lung RUL	3.39	3.47	0.27	0.079
14	52	Lung RUL	4.04	3.61	0.50	0.126
14	53	Lung RUL	2.75	3.48	0.25	0.093

15	54	Lung RLL	13.51	4.11	1.28	0.095
15	55	Lung RLL	9.83	3.26	0.97	0.099
15	56	Lung RLL	14.52	3.59	1.24	0.085
16	57	Internal mammary nodes	2.16	3.76	0.33	0.154
16	58	Internal mammary nodes	1.65	3.72	0.30	0.186
16	59	Internal mammary nodes	1.42	3.60	0.27	0.192
16	60	Internal mammary nodes	1.03	3.93	0.12	0.120
16	61	Internal mammary nodes	1.76	3.79	0.26	0.149
17	62	Pancreas	8.22	4.70	0.77	0.094
17	63	Pancreas	10.28	4.69	0.87	0.085
17	64	Pancreas	7.98	4.18	0.92	0.116
18	65	Pancreas	5.98	4.51	0.72	0.121
19	66	Retroperitoneum	1.52	5.29	0.21	0.139
19	67	Retroperitoneum	0.60	5.18	0.05	0.086
19	68	Retroperitoneum	0.40	4.54	0.04	0.108
20	69	Lung RLL	11.92	2.68	1.46	0.122
20	70	Lung RLL	11.29	2.74	1.51	0.134
20	71	Lung RLL	9.73	2.65	1.28	0.132
21	72	Lung LUL	0.76	3.49	0.22	0.309
21	73	Lung LUL	0.89	3.88	0.21	0.245
21	74	Lung LUL	0.88	3.90	0.15	0.176
22	75	Lung LUL	8.35	4.17	1.38	0.166
22	76	Lung LUL	7.27	3.29	1.45	0.200
22	77	Lung LUL	10.60	3.50	1.91	0.182
23	78	Lung Hilum Left	5.20	4.13	0.45	0.087
23	79	Lung Hilum Left	4.25	5.29	0.42	0.100
23	80	Lung Hilum Left	5.08	5.23	0.53	0.108
24	81	Chest wall	1.57	3.30	0.27	0.174
24	82	Chest wall	1.19	3.39	0.12	0.101
24	83	Chest wall	1.83	2.97	0.36	0.198
24	84	Chest wall	1.65	3.09	0.17	0.101
24	85	Chest wall	1.62	3.24	0.21	0.128
25	86	Lung LUL	3.03	3.77	0.85	0.293
25	87	Lung LUL	3.24	3.50	0.72	0.234
25	88	Lung LUL	5.74	4.60	1.31	0.234
26	89	Pancreas	3.46	4.47	0.50	0.146
26	90	Pancreas	5.47	3.90	0.46	0.085
26	91	Pancreas	6.54	4.40	0.46	0.070
27	92	Lung RLL	7.51	4.05	0.59	0.079
27	93	Lung RLL	9.27	3.94	0.81	0.088
27	94	Lung RLL	10.51	3.98	0.74	0.070
28	95	Lung LUL	5.54	4.55	0.51	0.092
28	96	Lung LUL	4.57	4.45	0.82	0.185
28	97	Lung LUL	6.20	3.72	0.55	0.089
29	98	Lung Hilum	8.03	4.98	0.82	0.104
29	99	Lung LAP	7.99	4.92	0.47	0.059
29	100	Lung Hilum	13.24	5.19	0.58	0.044
29	101	Lung Hilum	7.38	5.56	0.74	0.101
29	102	Lung Hilum	7.84	5.68	0.50	0.064
30	103	Lung Hilum Right	2.39	3.61	0.23	0.096
30	104	Lung Hilum Right	2.59	3.77	0.31	0.122
30	105	Lung Hilum Right	3.06	3.91	0.25	0.083
30	106	Lung Hilum Right	2.41	5.00	0.41	0.170
30	107	Lung Hilum Right	2.98	3.83	0.33	0.113
31	108	Lung Apex Left	0.75	4.11	0.16	0.228
31	109	Lung Apex Left	0.33	3.79	0.12	0.357
31	110	Lung Apex Left	0.49	3.61	0.10	0.201
32	111	Lung RLL	3.79	2.09	1.13	0.301
32	112	Lung RLL	2.23	2.01	0.60	0.277
32	113	Lung RLL	3.07	1.86	0.89	0.295
33	114	Lung LLL	9.49	3.79	1.18	0.125
33	115	Lung LLL	5.38	3.85	1.09	0.207
33	116	Lung LLL	7.16	3.88	1.30	0.182
33	117	Lung LLL	8.69	4.34	1.28	0.151

33	118	Lung LLL	7.53	4.38	1.21	0.195
34	119	Lung LUL	2.55	3.89	0.20	0.077
34	120	Lung LUL	4.40	3.61	0.54	0.123
34	121	Lung LUL	2.77	3.63	0.31	0.114
35	122	Lung RUL	2.92	3.75	0.24	0.084
35	123	Lung RUL	2.39	4.14	0.36	0.154
35	124	Lung RUL	3.70	3.28	0.36	0.098
36	125	Liver	5.95	2.63	0.60	0.101
36	126	Retroperitoneum	2.25	3.98	0.34	0.152
36	127	Liver	6.27	3.57	0.77	0.123
36	128	Retroperitoneum	1.57	2.50	0.57	0.369
36	129	Liver	6.22	3.06	0.69	0.111
37	130	Lung LUL	2.05	2.93	0.64	0.339
37	131	Lung LUL	4.21	2.86	0.38	0.090
37	132	Lung LUL	6.41	3.28	0.56	0.088
38	133	Chest wall	1.11	2.82	0.37	0.346
39	134	Pancreas	5.43	3.23	0.69	0.128
39	135	Pancreas	4.34	2.59	0.92	0.213
40	136	Lung LUL	5.40	3.27	1.13	0.212
40	137	Lung LUL	1.13	4.65	0.40	0.362
40	138	Lung LUL	2.50	4.77	0.65	0.270
41	139	Pancreas	2.84	3.65	0.41	0.147
41	140	Pancreas	2.62	3.56	0.39	0.151
41	141	Pancreas	4.08	3.69	0.48	0.119
42	142	Lung RUL	3.94	2.01	1.36	0.365
42	143	Lung RUL	5.44	2.72	1.58	0.295
42	144	Lung RUL	2.93	2.57	0.92	0.323
43	145	Lung LLL	6.32	3.30	0.46	0.074
43	146	Lung LLL	10.37	4.23	0.99	0.097
43	147	Lung LLL	4.66	3.43	0.86	0.187
43	148	Lung LLL	5.70	3.17	0.55	0.096
43	149	Lung LLL	9.24	3.72	1.11	0.121
44	150	Liver	11.60	4.74	0.98	0.085
44	151	Liver	9.63	4.96	0.82	0.085
44	152	Liver	9.35	5.14	0.53	0.056
45	153	Pancreas	9.70	3.63	0.88	0.091
45	154	Pancreas	10.32	3.88	0.74	0.072
45	155	Pancreas	7.23	3.25	0.66	0.091
46	156	Pancreas	5.83	6.15	0.60	0.103
46	157	Pancreas	4.86	6.34	0.73	0.152
46	158	Pancreas	6.99	6.60	0.53	0.076
		Average:	**4.96**	**3.88**	**0.60**	**0.141**

Table 3.2: Tumor and tumor motion characteristics for the fractions used in this study, and the results of prediction with 25-member LSTM-RNN ensembles at $t_{acq} = 280$ ms, $t_{sys} = 320$ ms, and a 30 s intermittent retraining interval. Bolded fractions are included in the optimization subset.

Fraction	Tumor Site	$\bar{A}$ (mm)	Mean Period (s)	ϕ	N_{HL}	w_{HL}	α_i	N_{seq}	$\tilde{C}_{treat}$, Fraction -Specific	$\tilde{C}_{treat}$, Global
1	Lung Hilum Right	6.95	2.63	LReLU	1	10	0.01	2 resp	0.20	0.20
4	Pancreas	7.74	3.99	sigmoid	1	50	0.1	1/2 resp	0.09	0.13
7	Lung RML	3.11	2.99	sigmoid	3	20	0.1	2 resp	0.12	0.14
11	Pancreas	2.83	3.92	LReLU	2	50	0.01	1/2 resp	0.18	0.21
17	Lung LUL	4.82	4.14	LReLU	2	20	0.01	1 resp	0.08	0.08
24	Lung LUL	6.72	3.49	ReLU	1	50	0.01	1 resp	0.12	0.14
26	Retroperitoneum	2.29	4.34	sigmoid	3	50	0.1	2 resp	0.15	0.16
31	Lung LUL	2.69	4.10	LReLU	2	20	0.01	1/2 resp	0.16	0.17
34	Lung RUL	4.28	4.08	tanh	1	50	0.01	1 resp	0.10	0.11
37	Lung LUL	3.95	3.01	sigmoid	3	20	0.01	2 resp	0.13	0.14
								Average:	**0.13**	**0.14**

Table 3.3: The optimal hyperparameter configurations for each fraction in the optimization set, and their amplitude-normalized predictive accuracy compared to that of the global optimal hyperparameter configuration.

ϕ	N_{HL}	w_{HL}	α_i	N_{seq}	Mean $\tilde{C}_{treat}$
LReLU	1	50	0.01	3.6 s	0.14
LReLU	1	20	0.01	1 resp	0.14
LReLU	1	50	0.01	1 resp	0.14
LReLU	1	20	0.01	3.6 s	0.15
ReLU	1	50	0.01	3.6 s	0.15
ReLU	1	20	0.01	3.6 s	0.15
ReLU	1	50	0.01	1/2 resp	0.15
LReLU	2	20	0.01	3.6 s	0.15
LReLU	1	50	0.01	1/2 resp	0.15
LReLU	2	50	0.01	3.6 s	0.15

Table 3.4: The 10 best-performing global hyperparameter settings, and their mean amplitude-normalized accuracy over the optimization set.

		t_{sys} (ms)											
		120	160	200	240	280	320	360	400	440	480	520	Average $\tilde{C}_{treat}$
t_{acq} (ms)	120	0.100	0.100	0.110	0.120	0.128	0.138	0.148	0.157	0.165	0.174	0.182	0.138
	160	0.089	0.095	0.105	0.115	0.125	0.136	0.144	0.153	0.162	0.171	0.178	0.134
	200	0.085	0.093	0.106	0.116	0.125	0.135	0.145	0.154	0.162	0.169	0.177	0.133
	240	0.082	0.089	0.105	0.118	0.127	0.135	0.144	0.153	0.163	0.172	0.179	0.133
	280	0.078	0.085	0.117	0.120	0.147	0.141	0.147	0.154	0.162	0.171	0.180	0.137

Table 3.5: $\widetilde{C}_{treat}$ as a function of varying t_{acq} and t_{sys} for 25-member LSTM-RNN ensembles and IR at 30 s intervals. The average $\widetilde{C}_{treat}$ taken across all t_{sys} (final column) shows little variation with t_{acq}.

Chapter 4

Conclusion

Dynamic tumour-tracked radiotherapy offers a means for greatly improving the conformality of EBRT treatments of highly-mobile tumours, potentially improving outcomes for a range of solid tumour types. Linac-MR based treatments are especially promising, owing to the unparalleled soft tissue contrast of MR (allowing for direct, markerless visualization of the tumour) and the lack of imaging dose concerns. However, inherently long system delays in MR-based tumour tracking have been challenging to address.

In Chapter 1 of this document, the basic ideas behind EBRT were outlined, with an emphasis on the concept of treatment conformality. Specific attention was paid to methods for compensating for intrafractional motion, both conventional and cutting-edge. Of particular interest was dynamic tumour-tracked EBRT performed on hybrid linac-MR systems, also known as nifteRT. The role of tumour motion prediction in nifteRT was described, and the strengths and limitations of previous work on this topic was discussed.

Chapter 2 described theoretical concepts that are important to understanding the work presented in this thesis. First, a thorough description of the construction, training and optimization of artificial neural networks was presented. Challenges associated with small training datasets (like those encountered in fraction-specific motion prediction) were identified, and methods for mitigat-

ing their effects were introduced. Next, the specific kind of neural networks used in this thesis, long short-term memory recurrent neural networks, were presented. There was then a brief discussion on network adaptation strategies, before moving on to the theory of MRI. The focus of this section was to introduce the main determinants of MR acquisition time, a major contributor to system delay in linac-MR based tumour tracking. Some strategies for reducing acquisition time were then discussed.

Chapter 3 contained the main scientific contribution of this thesis, a novel method for training neural networks for respiration-induced tumour motion prediction. Three major novelties were introduced: (1) super-convergence regularization paired with intelligent early stopping to get training times down to < 5 seconds per network on non-specialized hardware; (2) homogeneous network ensembles to improve the accuracy and stability of the predictions at no cost to training time; and (3) a novel adaptation approach that improved the rapidity and depth of the networks' response to intrafractional changes in respiratory patterns.

The initial questions that this research was intended to answer were outlined at the end of Section 1.12. With respect to the question of an an optimal HPO strategy, it was found that patient- and fraction-specific HPO would yield minimal improvements to predictive accuracy compared to using a universal set of network hyperparameters, especially when the computational expense of HPO is considered. The universally optimized hyperparameters identified in Chapter 3 not only provide good predictive accuracy, but also extremely short training times that yield significant benefits in terms of adaptation to changing patient respiratory patterns.

It was found in Chapter 3 that predictive accuracy is largely independent of acquisition rate in the range of 120 – 280 ms. Unsurprisingly, the magnitude of the system delay was found to strongly affect predictive accuracy, increasing

from 9% to 18% of the mean tumour amplitude when system delay ranges from 120 ms to 520 ms. There is therefore significant motivation to decrease MR acquisition times not because it would allow for a finer temporal sampling of the tumour motion, but because it would result in shorter acquisition-related system delays.

Interestingly, the compact universally optimal network architecture determined in Chapter 3 removed some of the motivation for experimenting with GPU acceleration. GPUs are useful tools for training deeper, more complex networks since they more efficiently handle large volumes of computations that can be performed in parallel. However, it takes time to transfer data to and from the GPU, and in the case of these compact networks the time lost to data transfer was significantly larger than the time saved by performing the relatively few required computations in a more efficient manner.

Chapter 3 envisioned a potential clinical application of the approach developed in this thesis. First, a patient's respiration would be observed for two minutes to establish training and validation sets. There would then be a few seconds of delay while the ensemble of networks were trained on these data, after which treatment would commence. During treatment, a new ensemble of networks would be training in the background using the data that were acquired during the initial training session. This process of training, substitution of a new predictor and initialization of a new training session in the background would continue for as long as the treatment lasts, resulting in a 100% duty cycle and predictors that are always based off of the most recent (and therefore most relevant) motion data.

There are several directions in which I think this research project could proceed. First, it should be tested in real-time on real hardware, such as the Alberta Linac-MR. Initial experiments could focus only on tracking accuracy for a phantom programmed to mimic real respiratory motion traces, but later

experiments should also analyze the dosimetric accuracy of a treatment delivered with real-time tracking. The problem of how to detect tracking failures and trigger appropriate halting mechanisms also needs to be explored. Finally, a method for predicting tracking accuracy during treatment for dynamic adjustment of the treatment margins could be investigated. It is reasonable to suppose that the validation cost function measured during a training session might correlate to the predictive accuracy of that network thereafter, in which case margins could be enlarged or reduced in anticipation of future performance. This may prove to be impractical compared to setting static margins, but it may be worth exploring.

References

[1] D. Brenner, A. Poirer, R. Woods, *et al.*, "Projected estimates of cancer in Canada in 2022," *C.M.A.J.*, vol. 194, E601–E607, 2022. DOI: 10.1503/cmaj.212097. 2

[2] *Radiation Therapy, Canadian Cancer Society*, https://web.archive.org/web/20220204151511/https://cancer.ca/en/treatments/treatment-types/radiation-therapy, Accessed: 2022-06-02. 2

[3] M. B. Barton, M. Frommer, and J. Shafiq, "Role of radiotherapy in cancer control in low-income and middle-income countries," *Lancet Oncol.*, vol. 7, pp. 584–595, 2006. DOI: 10.1016/S1470-2045(06)70759-8. 2

[4] E. Weiss, M. Fatyga, Y. Wu, *et al.*, "Dose escalation for locally advanced lung cancer using adaptive radiation therapy with simultaneous integrated volume-adapted boost," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 86, pp. 414–419, 2013. DOI: 10.1016/j.ijrobp.2012.12.027. 3

[5] H. Park, J. Seong, K. Han, C. Chon, Y. Moon, and C. Suh, "Dose-response relationship in local radiotherapy for hepatocellular carcinoma," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 54, pp. 150–155, 2002. DOI: 10.1016/S0360-3016(02)02864-X. 3

[6] M. Velec, J. L. Moseley, L. A. Dawson, and K. K. Brock, "Dose escalated liver stereotactic body radiation therapy at the mean respiratory position," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 89, 2014. DOI: 10.1016/J.IJROBP.2014.04.051. 3

[7] C. H. Chapman, C. McGuinness, A. R. Gottschalk, *et al.*, "Influence of respiratory motion management technique on radiation pneumonitis risk with robotic stereotactic body radiation therapy," *J. Appl. Clin. Med. Phys.*, vol. 19, pp. 48–57, 2018. DOI: 10.1002/acm2.12338. 3

[8] E. Wolny-Rokicka, A. Tukiendorf, J. Wydmański, D. Roszkowska, B. Staniul, and A. Zembroń-Łacny, "Thyroid function after postoperative radiation therapy in patients with breast cancer," *Asian Pac. J. Cancer. Prev.*, vol. 17, 2016. DOI: 10.22034/apjcp.2016.17.10.4577. 3

[9] International Commission on Radiological Units and Measurements, "ICRU Report 50. Prescribing, recording, and reporting photon beam therapy," *Journal of the ICRU*, vol. os-26, pp. 1–72, 1993. 3

[10] International Commission on Radiological Units and Measurements, "ICRU Report 62. Prescribing, recording, and reporting photon beam therapy (Supplement to ICRU Report 50)," *Journal of the ICRU*, vol. os-32, pp. 1–52, 1999. 3, 5, 76, 117

[11] International Commission on Radiological Units and Measurements, "ICRU Report 83. Prescribing, recording, and reporting intensity-modulated photon-beam therapy (IMRT)," *Journal of the ICRU*, vol. 10, pp. 1–106, 2010. 3

[12] J. Stroom, H. de Boer, H. Huizenga, and A. Visser, "Inclusion of geometrical uncertainties in radiotherapy treatment planning by means of coverage probability," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 43, pp. 905–919, 1999. DOI: 10.1016/s0360-3016(98)00468-4. 6

[13] M. van Herk, C. Remeijer, C. Rasch, and J. Lebesque, "The probability of correct target dosage: Dose-population histograms for deriving treatment margins in radiotherapy," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 47, pp. 1121–1135, 2000. DOI: 10.1016/s0360-3016(00)00518-6. 6

[14] L. J. Verhey, "Immobilizing and positioning patients for radiotherapy," *Semin. Radiat. Oncol.*, vol. 5, pp. 100–114, 1995. DOI: 10.1016/S1053-4296(95)80004-2. 7

[15] R. Jadon, C. Pembroke, C. Hanna, *et al.*, "A systematic review of organ motion and image-guided strategies in external beam radiotherapy for cervical cancer," *Clin. Oncol. (R. Coll. Radiol.)*, vol. 26, pp. 185–196, 2014. DOI: 10.1016/j.clon.2013.11.031. 7

[16] S. Heng, S. Low, and K. Sivamany, "The influence of the bowel and bladder preparation protocol for radiotherapy of prostate cancer using kilo-voltage cone beam CT: Our experience," *Indian J Cancer*, vol. 52, pp. 639–644, 2015. 7

[17] D. Yan, F. Vicini, J. Wong, and A. Martinez, "Adaptive radiation therapy," *Phys. Med. Biol.*, vol. 42, pp. 123–132, 1997. DOI: 10.1088/0031-9155/42/1/008. 7

[18] D. Verellen, M. De Ridder, and G. Storme, "A (short) history of image-guided radiotherapy," *Radiother. Oncol.*, vol. 86, pp. 4–13, 2008. DOI: 10.1016/j.radonc.2007.11.023. 8

[19] S. Camps, D. Fontanarosa, P. de With, F. Verhaegen, and B. Vanneste, "The use of ultrasound imaging in the external beam radiotherapy workflow of prostate cancer patients," *Biomed. Res. Int.*, vol. 2018, p. 7 569 590, 2018. DOI: 10.1155/2018/7569590. 9

[20] K. Singh, K. Bønaa, S. Solberg, D. Sørlie, and L. Bjørk, "Intra- and interobserver variability in ultrasound measurements of abdominal aortic diameter. The Tromsø study," *Eur. J. Vasc. Endovasc. Surg.*, vol. 15, pp. 497–504, 1998. DOI: 10.1016/s1078-5884(98)80109-3. 9

[21] M. Fargier-Voiron, B. Presles, P. Pommier, *et al.*, "Impact of probe pressure variability on prostate localization for ultrasound-based image-guided radiotherapy," *Radiother. Oncol.*, vol. 111, pp. 132–137, 2014. DOI: 10.1016/j.radonc.2014.02.008. 9

[22] C. Western, D. Hristov, and J. Schlosser, "Ultrasound imaging in radiation therapy: From interfractional to intrafractional guidance," *Cureus*, vol. 7, e280, 2015. DOI: 10.7759/cureus.280. 9

[23] M. van Vulpen, C. Field, C. P. Raaijmakers, *et al.*, "Comparing step-and-shoot IMRT with dynamic helical tomotherapy IMRT plans for head-and-neck cancer," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 62, pp. 1535–1539, 2005. DOI: https://doi.org/10.1016/j.ijrobp.2005.04.011. 9

[24] B. G. Fallone, D. R. C. Rivest, T. A. Riauka, and A. D. Murtha, "Assessment of a commercially available automatic deformable registration system," *J. Appl. Clin. Med. Phys.*, vol. 11, pp. 101–123, 2010. DOI: https://doi.org/10.1120/jacmp.v11i3.3175. 9

[25] D. Jaffray, M. Carlone, M. Milosevic, *et al.*, "A facility for magnetic resonance-guided radiation therapy," *Semin. Radiat. Oncol.*, vol. 24, pp. 193–195, 2014. DOI: 10.1016/j.semradonc.2014.02.012. 9

[26] B. G. Fallone, "The rotating biplanar linac-magnetic resonance imaging system," *Semin. Radiat. Oncol.*, vol. 24, pp. 200–202, 2014. DOI: 10.1016/j.semradonc.2014.02.011. 9, 15, 72

[27] J. Lagendijk, B. Raymakers, and M. van Vulpen, "The magnetic resonance imaging-linac system," *Semin. Radiat. Oncol.*, vol. 24, pp. 207–209, 2014. DOI: 10.1016/j.semradonc.2014.02.009. 9

[28] S. Mutic and J. Dempsey, "The viewRay system: Magnetic resonance–guided and controlled radiotherapy," *Semin. Radiat. Oncol.*, vol. 24, pp. 196–199, 2014. DOI: 10.1016/j.semradonc.2014.02.008. 9

[29] B. G. Fallone, B. Murray, S. Rathee, *et al.*, "First MR images obtained during megavoltage photon irradiation from a prototype integrated linac-MR system," *Med. Phys.*, vol. 36, pp. 2084–2088, 2009. DOI: 10.1118/1.3125662. 9, 72

[30] *The Management of Respiratory Motion in Radiation Oncology: Report of AAPM Task Group 76*. College Park, MD, USA: American Association of Physicists in Medicine, 2006. DOI: 10.1118/1.2349696. 10, 13, 68, 72

[31] L. Giuranno, J. Ient, D. De Ruysscher, and M. Vooijs, "Radiation-induced lung injury (RILI)," *Front. Oncol.*, vol. 9, p. 877, 2019. DOI: doi:10.3389/fonc.2019.00877. 11

[32] K. E. Rosenzweig, J. Hanley, D. Mah, *et al.*, "The deep inspiration breath-hold technique in the treatment of inoperable non–small-cell lung cancer," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 48, pp. 81–87, 2000. DOI: 10.1016/s0360-3016(00)00583-6. 12

[33] V. M. Remouchamps, F. A. Vicini, M. B. Sharpe, L. L. Kestin, A. A. Martinez, and J. W. Wong, "Significant reductions in heart and lung doses using deep inspiration breath hold with active breathing control and intensity-modulated radiation therapy for patients treated with locoregional breast irradiation," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 55, pp. 392–406, 2003. DOI: 10.1016/s0360-3016(02)04143-3. 12

[34] Y. Negoro, Y. Nagata, T. Aoki, *et al.*, "The effectiveness of an immobilization device in conformal radiotherapy for lung tumor: Reduction of respiratory tumor movement and evaluation of the daily setup accuracy," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 50, pp. 889–898, 2001. DOI: 10.1016/s0360-3016(01)01516-4. 12

[35] P. Keall, "4-dimensional computed tomography imaging and treatment planning," vol. 14, pp. 81–90, 2004. DOI: 10.1053/j.semradonc.2003.10.006. 12

[36] D. P. Gierga, J. Brewer, G. C. Sharp, M. Betke, C. G. Willett, and G. T. Chen, "The correlation between internal and external markers for abdominal tumors: Implications for respiratory gating," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 61, pp. 1551–1558, 2005. DOI: 10.1016/j.ijrobp.2004.12.013. 13

[37] A. Sawant, R. Venkat, V. Srivastava, *et al.*, "Management of three-dimensional intrafraction motion through real-time DMLC tracking," *Med. Phys.*, vol. 35, pp. 2050–2061, 2008. DOI: 10.1118/1.2905355. 14, 72

[38] P. J. Keall, H. Cattell, D. Pokhrel, *et al.*, "Geometric accuracy of a real-time target tracking system with dynamic multileaf collimator tracking system," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 65, pp. 1579–1584, 2006. DOI: 10.1016/j.ijrobp.2006.04.038. 14, 72

[39] D. McQuaid, M. Partridge, J. Symonds-Tayler, P. Evans, and S. Webb, "Target-tracking deliveries on an Elekta linac: A feasibility study," *Phys. Med. Biol.*, vol. 54, p. 3563, 2009. DOI: 10.1088/0031-9155/54/11/019. 14, 72

[40] M. B. Tacke, S. Nill, A. Krauss, and U. Oelfke, "Real-time tumor tracking: Automatic compensation of target motion using the siemens 160 MLC," *Med. Phys.*, vol. 37, pp. 753–761, 2010. DOI: 10.1118/1.3284543. 14, 72

[41] A. Krauss, S. Nill, M. Tacke, and U. Oelfke, "Electromagnetic real-time tumor position monitoring and dynamic multileaf collimator tracking using a Siemens 160 MLC: Geometric and dosimetric accuracy of an integrated system," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 79, pp. 579–587, 2011. DOI: 10.1016/j.ijrobp.2010.03.043. 14, 72

[42] T. Depuydt, D. Verellen, O. Haas, *et al.*, "Geometric accuracy of a novel gimbals based radiation therapy tumor tracking system," *Radiother. Oncol.*, vol. 98, pp. 365–372, 2011. DOI: 10.1016/j.radonc.2011.01.015. 14, 72

[43] J. R. Adler Jr, S. D. Chang, M. J. Murphy, J. Doty, P. Geis, and S. L. Hancock, "The Cyberknife: A frameless robotic system for radiosurgery," *Stereotact. Funct. Neurosurg.*, vol. 69, pp. 124–128, 1997. DOI: 10.1159/000099863. 14, 72

[44] W. D D'Souza, S. A. Naqvi, and X. Y. Cedric, "Real-time intra-fraction-motion tracking using the treatment couch: A feasibility study," *Phys. Med. Biol.*, vol. 50, p. 4021, 2005. DOI: 10.1088/0031-9155/50/17/007. 14, 72

[45] R. Li and G. Sharp, "Robust fluoroscopic tracking of fiducial markers: Exploiting the spatial constraints," *Phys. Med. Biol.*, vol. 58, p. 1789, 2013. DOI: 10.1088/0031-9155/58/6/1789. 14

[46] S. Dieterich, D. Taylor, C. Chuang, K. Wong, J. Tang, and K. Main, "The CyberKnife Synchrony respiratory tracking system: Evaluation of systematic targeting uncertainty," *Sunnyvale, CA: Accuray Inc*, 2004. 14

[47] J. M. Balter, J. N. Wright, L. J. Newell, *et al.*, "Accuracy of a wireless localization system for radiotherapy," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 61, pp. 933–937, 2005. DOI: 10.1016/j.ijrobp.2004.11.009. 14

[48] N. Kothary, J. J. Heit, J. D. Louie, *et al.*, "Safety and efficacy of percutaneous fiducial marker implantation for image-guided radiation therapy," *J. Vasc. Interv. Radiol.*, vol. 20, pp. 235–239, 2009. DOI: 10.1016/j.jvir.2008.09.026. 14

[49] K. Wachowicz, N. De Zanche, E. Yip, V. Volotovskyy, and B. G. Fallone, "CNR considerations for rapid real-time MRI tumor tracking in radiotherapy hybrid devices: Effects of B0 field strength," *Med. Phys.*, vol. 43, pp. 4903–4914, 2016. DOI: https://doi.org/10.1118/1.4959542. 15

[50] J. Yun, S. Rathee, and B. G. Fallone, "A deep-learning based 3d tumor motion prediction algorithm for non-invasive intra-fractional tumor-tracked radiotherapy (nifteRT) on linac-MR," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 105, S28, 2019. DOI: 10.1016/j.ijrobp.2019.06.434. 15, 18, 19, 72–74, 96

[51] J. Yun, K. Wachowicz, M. Mackenzie, S. Rathee, D. Robinson, and B. G. Fallone, "First demonstration of intrafractional tumor-tracked irradiation using 2D phantom MR images on a prototype linac-MR," *Med. Phys.*, vol. 40, p. 051 718, 2013. DOI: 10.1118/1.4802735. 16, 72

[52] E. Yip, J. Yun, Z. Gabos, *et al.*, "Evaluating performance of a user-trained MR lung tumor autocontouring algorithm in the context of intra- and interobserver variations," *Med. Phys.*, vol. 45, pp. 307–313, 2018. DOI: https://doi.org/10.1002/mp.12687. 17

[53] M. J. Murphy, M. Isaakson, and J. Jalden, "Adaptive filtering to predict lung tumor motion during free breathing," in *CARS 2002 computer assisted radiology and surgery*, Springer, 2002, pp. 539–544. DOI: 10.1007/978-3-642-56168-9_90. 17, 72

[54] G. C. Sharp, S. B. Jiang, S. Shimizu, and H. Shirato, "Prediction of respiratory tumour motion for real-time image-guided radiotherapy," *Phys. Med. Biol.*, vol. 49, p. 425, 2004. DOI: 10.1088/0031-9155/49/3/006. 17, 72

[55] S. Vedam, P. Keall, A. Docef, D. Todor, V. Kini, and R. Mohan, "Predicting respiratory motion for four-dimensional radiotherapy," *Med. Phys.*, vol. 31, pp. 2274–2283, 2004. DOI: 10.1118/1.1771931. 17, 72

[56] M. Isaksson, J. Jalden, and M. J. Murphy, "On using an adaptive neural network to predict lung tumor motion during respiration for radiotherapy applications," *Med. Phys.*, vol. 32, pp. 3801–3809, 2005. DOI: 10.1118/1.1771931. 17, 72

[57] Q. Ren, S. Nishioka, H. Shirato, and R. I. Berbeco, "Adaptive prediction of respiratory motion for motion compensation radiotherapy," *Phys. Med. Biol.*, vol. 52, p. 6651, 2007. DOI: 10.1088/0031-9155/52/22/007. 17

[58] F. Ernst, A. Schlaefer, and A. Schweikard, "Prediction of respiratory motion with wavelet-based multiscale autoregression," in *International conference on medical image computing and computer-assisted intervention*, Springer, 2007, pp. 668–675. DOI: 10.1007/978-3-540-75759-7_81. 17

[59] K. McCall and R. Jeraj, "Dual-component model of respiratory motion based on the periodic autoregressive moving average (periodic ARMA) method," *Phys. Med. Biol.*, vol. 52, p. 3455, 2007. DOI: 10.1088/0031-9155/52/12/009. 17

[60] F. Ernst, A. Schlaefer, S. Dieterich, and A. Schweikard, "A fast lane approach to LMS prediction of respiratory motion signals," *Biomed. Signal Process. Control*, vol. 3, pp. 291–299, 2008. DOI: 10.1016/j.bspc.2008.06.001. 17

[61] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015. DOI: 10.1038/nature14539. 17

[62] J. H. Goodband, O. C. Haas, and J. Mills, "A comparison of neural network approaches for on-line prediction in IGRT," *Med. Phys.*, vol. 35, pp. 1113–1122, 2008. DOI: 10.1118/1.2836416. 18

[63] M. J. Murphy and D. Pokhrel, "Optimization of an adaptive neural network to predict breathing," *Med. Phys.*, vol. 36, pp. 40–47, 2009. DOI: 10.1118/1.3026608. 18, 72

[64] P. Verma, H. Wu, M. Langer, I. Das, and G. Sandison, "Survey: Real-time tumor motion prediction for image-guided radiation treatment," *Comput. Sci. Eng.*, vol. 13, pp. 24–35, 2010. DOI: 10.1109/MCSE.2010.99. 18, 73

[65] I. Buzurovic, K. Huang, T. K. Podder, and Y. Yu, "Comparison between acceleration-enhanced adaptive filters and neural network filters for respiratory motion prediction," in *11th Symposium on Neural Network Applications in Electrical Engineering*, IEEE, 2012, pp. 181–184. DOI: 10.1109/NEUREL.2012.6420003. 18

[66] J. Yun, M. Mackenzie, S. Rathee, D. Robinson, and B. G. Fallone, "An artificial neural network (ANN)-based lung-tumor motion predictor for intrafractional MR tumor tracking," *Med. Phys.*, vol. 39, pp. 4423–4433, 2012. DOI: 10.1118/1.4730294. 18, 72, 74, 92

[67] S. J. Lee, Y. Motai, and M. Murphy, "Respiratory motion estimation with hybrid implementation of extended Kalman filter," *IEEE Trans. Ind. Electron.*, vol. 59, pp. 4421–4432, 2011. DOI: 10.1109/TIE.2011.2158046. 18, 72

[68] B.-H. Jung, B.-H. Kim, and S.-M. Hong, "Respiratory motion prediction with extended Kalman filters based on local circular motion model," *Int. J. BioSci. Biotech.*, vol. 5, pp. 51–58, 2013. 18

[69] S. Choi, Y. Chang, N. Kim, S. H. Park, S. Y. Song, and H. S. Kang, "Performance enhancement of respiratory tumor motion prediction using adaptive support vector regression: Comparison with adaptive neural network method," *Int. J. Imaging Syst. Technol.*, vol. 24, pp. 8–15, 2014. DOI: 10.1002/ima.22073. 18, 72

[70] S. Hong and W. Bukhari, "Real-time prediction of respiratory motion using a cascade structure of an extended Kalman filter and support vector regression," *Phys. Med. Biol.*, vol. 59, p. 3555, 2014. DOI: 10.1088/0031-9155/59/13/3555. 18, 72

[71] R. Dürichen, T. Wissel, F. Ernst, and A. Schweikard, "Respiratory motion compensation with relevance vector machines," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2013, pp. 108–115. DOI: 10.1007/978-3-642-40763-5_14. 18, 72

[72] H. Lin, C. Shi, B. Wang, M. F. Chan, X. Tang, and W. Ji, "Towards real-time respiratory motion prediction based on long short-term memory neural networks," *Phys. Med. Biol.*, vol. 64, p. 085 010, 2019. DOI: 10.1088/1361-6560/ab13fa. 18, 72

[73] P. Chang, J. Dang, J. Dai, W. Sun, *et al.*, "Real-time respiratory tumor motion prediction based on a temporal convolutional neural network: Prediction model development study," *J. Med. Internet Res.*, vol. 23, e27235, 2021. DOI: 10.2196/27235. 18

[74] S. Herculano-Houzel, "The human brain in numbers: A linearly scaled-up primate brain," *Front. Hum. Neurosci.*, vol. 3, p. 31, 2009. DOI: 10.3389/neuro.09.031.2009. 22

[75] R. von Bernhardi, L. Eugenín-von Bernhardi, and J. Eugenín, "What is neural plasticity?" *Adv. Exp. Med. Biol.*, vol. 1015, pp. 1–15, 2017. DOI: 10.1007/978-3-319-62817-2_1. 22

[76] L. F. Abbott and S. B. Nelson, "Synaptic plasticity: Taming the beast," *Nat. Neurosci.*, vol. 3 Suppl, pp. 1178–1183, 2000. DOI: 10.1038/81453. 22

[77] L. Floridi and M. Chiriatti, "GPT-3: Its nature, scope, limits, and consequences," *Minds Mach. (Dordr.)*, vol. 30, pp. 681–694, 2020. DOI: 10.1007/s11023-020-09548-1. 36

[78] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, 2012. DOI: 10.5555/2188385.2188395. 38

[79] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, IEEE, 1995. DOI: 10.1109/ICNN.1995.488968. 39

[80] N. Hansen, S. D. Müller, and P. Koumoutsakos, "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)," *Evol. Comput.*, vol. 11, pp. 1–18, 2003. DOI: 10.1162/106365603321828970. 39

[81] T. Dietterich, "Overfitting and undercomputing in machine learning," *ACM Comput. Surv.*, vol. 27, pp. 326–327, 1995. DOI: 10.1145/212094.212114. 41

[82] A. Alwosheel, S. van Cranenburgh, and C. G. Chorus, "Is your dataset big enough? Sample size requirements when using artificial neural networks for discrete choice analysis," *J. Choice Model.*, vol. 28, pp. 167–182, 2018. DOI: 10.1016/j.jocm.2018.07.002. 43

[83] P. Baldi and P. J. Sadowski, "Understanding dropout," in *Advances in Neural Information Processing Systems*, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds., vol. 26, Curran Associates, Inc., 2013. 43

[84] L. N. Smith and N. Topin, "Super-convergence: Very fast training of neural networks using large learning rates," in *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, T. Pham, Ed., SPIE, 2019. DOI: 10.1117/12.2520589. 45, 74

[85] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, pp. 1735–1780, 1997. DOI: 10.1162/neco.1997.9.8.1735. 51

[86] *The most cited neural networks all build on work done in my labs*, https://people.idsia.ch/~juergen/most-cited-neural-nets.html, Accessed: 2022-07-18. 53

[87] H. Y. Carr, "Steady-state free precession in nuclear magnetic resonance," *Phys. Rev.*, vol. 112, pp. 1693–1701, 1958. DOI: 10.1103/PhysRev.112.1693. 66

[88] O. Bieri and K. Scheffler, "Fundamentals of balanced steady state free precession MRI," *J. Magn. Reson. Imaging*, vol. 38, pp. 2–11, 2013. DOI: 10.1002/jmri.24163. 67

[89] M. Uecker, S. Zhang, D. Voit, A. Karaus, K.-D. Merboldt, and J. Frahm, "Real-time MRI at a resolution of 20 ms," *NMR Biomed.*, vol. 23, pp. 986–994, 2010. DOI: 10.1002/nbm.1585. 68, 69

[90] L. Feng, N. Tyagi, and R. Otazo, "MRSIGMA: Magnetic resonance SIGnature MAtching for real-time volumetric imaging," *Magn. Reson. Med.*, vol. 84, pp. 1280–1292, 2020. DOI: 10.1002/mrm.28200. 69

[91] A. Krauss, S. Nill, and U. Oelfke, "The comparative performance of four respiratory motion predictors for real-time tumour tracking," *Phys. Med. Biol.*, vol. 56, pp. 5303–5317, 2011. DOI: 10.1088/0031-9155/56/16/015. 72, 73

[92] A. Jöhl, S. Ehrbar, M. Guckenberger, *et al.*, "Performance comparison of prediction filters for respiratory motion tracking in radiotherapy," *Med. Phys.*, vol. 47, pp. 643–650, 2020. DOI: 10.1002/mp.13929. 72, 73

[93] F. Ernst, R. Dürichen, A. Schlaefer, and A. Schweikard, "Evaluating and comparing algorithms for respiratory motion prediction," *Phys. Med. Biol.*, vol. 58, pp. 3911–3929, 2013. DOI: 10.1088/0031-9155/58/11/3911. 72, 73

[94] X. Li, Y.-H. Lee, S. Mikaiel, J. Simonelli, T.-C. Tsao, and H. H. Wu, "Respiratory motion prediction using fusion-based multi-rate kalman filtering and real-time golden-angle radial MRI," *IEEE Trans. Biomed. Eng.*, vol. 67, pp. 1727–1738, 2020. DOI: 10.1109/TBME.2019.2944803. 72

[95] W. Bukhari and S.-M. Hong, "Real-time prediction and gating of respiratory motion using an extended kalman filter and gaussian process regression," *Phys. Med. Biol.*, vol. 60, pp. 233–252, 2015. DOI: 10.1088/0031-9155/60/1/233. 72

[96] F. Ernst and A. Schweikard, "Forecasting respiratory motion with accurate online support vector regression (SVRpred)," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 4, pp. 439–447, 2009. DOI: 10.1007/s11548-009-0355-5. 72

[97] N. Riaz, P. Shanker, R. Wiersma, *et al.*, "Predicting respiratory tumor motion with multi-dimensional adaptive filters and support vector regression," *Phys. Med. Biol.*, vol. 54, pp. 5735–5748, 2009. DOI: 10.1088/0031-9155/54/19/005. 72

[98] Q. Fan, X. Yu, Y. Zhao, and S. Yu, "A respiratory motion prediction method based on improved relevance vector machine," *Mob. Netw. Appl.*, vol. 25, pp. 2270–2279, 2020. DOI: 10.1007/s11036-020-01610-7. 72

[99] M. Lee, M.-S. Cho, H. Lee, *et al.*, "Geometric and dosimetric verification of a recurrent neural network algorithm to compensate for respiratory motion using an articulated robotic couch," *J. Korean Phys. Soc.*, vol. 78, pp. 64–72, 2021. DOI: 10.1007/s40042-020-00013-x. 72, 73

[100] M. Mafi and S. M. Moghadam, "Real-time prediction of tumor motion using a dynamic neural network," *Med. Biol. Eng. Comput.*, vol. 58, pp. 529–539, 2020. DOI: 10.1007/s11517-019-02096-6. 72

[101] T. P. Teo, S. B. Ahmed, P. Kawalec, *et al.*, "Feasibility of predicting tumor motion using online data acquired during treatment and a generalized neural network optimized with offline patient tumor trajectories," *Med. Phys.*, vol. 45, pp. 830–845, 2018. DOI: 10.1002/mp.12731. 72

[102] M. J. Murphy and S. Dieterich, "Comparative performance of linear and nonlinear neural networks to predict irregular breathing," *Phys. Med. Biol.*, vol. 51, pp. 5903–5914, 2006. DOI: 10.1088/0031-9155/51/22/012. 73

[103] S. Nabavi, M. Abdoos, M. E. Moghaddam, and M. Mohammadi, "Respiratory motion prediction using deep convolutional long short-term memory network," *J. Med. Signals Sens.*, vol. 10, pp. 69–75, 2020. DOI: 10.4103/jmss.JMSS_38_19. 73

[104] A. Pasini, "Artificial neural networks for small dataset analysis," *J. Thorac. Dis.*, vol. 7, pp. 953–960, 2015. DOI: 10.3978/j.issn.2072-1439.2015.04.61. 73, 74

[105] Y. Grushka-Cockayne, V. R. R. Jose, and K. C. Lichtendahl Jr, "Ensembles of overfit and overconfident forecasts," *Manage. Sci.*, vol. 63, pp. 1110–1130, 2017. DOI: 10.1287/mnsc.2015.2389. 74, 83

[106] Y. Suh, S. Dieterich, B. Cho, and P. J. Keall, "An analysis of thoracic and abdominal tumour motion for stereotactic body radiotherapy patients," *Phys. Med. Biol.*, vol. 53, pp. 3623–3640, 2008. DOI: 10.1088/0031-9155/53/13/016. 74

[107] A. Gaya, P. Camilleri, A. Nash, D. Hughes, and J. Good, "Implementation of stereotactic MRI-guided adaptive radiotherapy (SMART) for hepatobiliary and pancreatic cancers in the united kingdom - fifty in five," *Cureus*, vol. 13, e15075, 2021. DOI: 10.7759/cureus.15075. 75

[108] M. Glitzner, P. L. Woodhead, P. T. S. Borman, J. J. W. Lagendijk, and B. W. Raaymakers, "Technical note: MLC-tracking performance on the elekta unity MRI-linac," *Phys. Med. Biol.*, vol. 64, 15NT02, 2019. DOI: 10.1088/1361-6560/ab2667. 75

[109] M. van Herk, "Errors and margins in radiotherapy," *Semin. Radiat. Oncol.*, vol. 14, pp. 52–64, 2004. DOI: 10.1053/j.semradonc.2003.10.003. 76

[110] J. C. Stroom and B. J. M. Heijmen, "Geometrical uncertainties, radiotherapy planning margins, and the ICRU-62 report," *Radiother. Oncol.*, vol. 64, pp. 75–83, 2002. DOI: 10.1016/s0167-8140(02)00140-8. 76

[111] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014. DOI: 10.48550/arXiv.1412.6980. 78

[112] L. N. Smith, "Cyclical learning rates for training neural networks," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Santa Rosa, CA, USA: IEEE, 2017. DOI: 10.1109/WACV.2017.58. 80, 86

[113] B. Pérez-Sánchez, O. Fontenla-Romero, and B. Guijarro-Berdiñas, "A review of adaptive online learning for artificial neural networks," *Artif. Intell. Rev.*, vol. 49, pp. 281–299, 2018. DOI: 10.1007/s10462-016-9526-2. 84

[114] G. Van Rossum and F. Drake Jr., *Python reference manual*. Amsterdam: Centrum voor Wiskunde en Informatica, 1995. 86

[115] M. Abadi, A. Agarwal, P. Barham, *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," 2016. DOI: 10.48550/arXiv.1603.04467. 86

[116] M. Wright, B. Dietz, E. Yip, *et al.*, "Time domain principal component analysis for rapid, real-time 2D MRI reconstruction from undersampled data," *Med. Phys.*, vol. 48, pp. 6724–6739, 2021. DOI: 10.1002/mp.15238. 97

[117] F. Ernst, A. Schlaefer, and A. Schweikard, "Predicting the outcome of respiratory motion prediction," *Med. Phys.*, vol. 38, pp. 5569–5581, 2011. DOI: 10.1118/1.3633907. 97

Appendix A

Copyright Transfer Information

Figure 1.1

This figure originally appeared as Figure 2.6 in ICRU 62[10].

Author: T. Landberg, J. Chavaudra, J. Dobbs, et al.

Publication: Journal of the ICRU

Publisher: SAGE Publications

Date: 11/01/1999