Applied Mathematical Modelling of Pulmonary Function Tests

I am not sure of the title. How do we also introduce the ML bit?

Cezar Trișcă-Vicol

Christ Church College

University of Oxford

Department of Computer Science
Supervisor: David Kay

A project report presented for
Third year MSc Computer Science

Trinity 2023

Acknowledgements

Firstly, to Professor David Kay, thank you for supervising this project and providing me with amazing guidance. I am deeply grateful for the excitement we shared for the results of our work.

Secondly, to Irwin Zaid. You helped me find and use the right resources for this task through countless meetings and discussion. I hope I didn't bother you too much with my unending need for assistance.

This work followed the footsteps of Brody Foy's thesis (2018)[1].

Abstract

The lungs are a complex organ affected by both shape and structure. In standard clinical practice, lung function is assessed through analysis of 'at the mouth' measurements, which apply a smoothing function on the state and structure of the lungs. This means that assessments are usually insensitive and oversimplify the actual behavior of the lungs. We are looking to develop a computational model that has the ability to, given more complex results to assessments such as multiple breath washout and magnetic resonance imaging, change its structure so as to emulate these same results. We would then be able to create critical clinical insight into how these test indices can be interpreted by health officials.

Contents

Chapter 1

Introduction

The human lung is a complex organ affected by both shape and structure. As a result, small changes in tissue structure or the size of certain airways can create observable changes in ventilation patterns. As a result, the accurate analysis of the lung structure and ventilation mechanics of specific patients is quite a challenge. This is made worse by the fact that most lung diseases are chronic and so treatment is more often than not focused on slowing the progression of the disease instead of symptom reversal. This creates a strong need for accurate systems able to detect morphological changes induced by disease.

1.1 Motivation

In most clinical settings, spirometry is used to test lung function, which evaluates flow rate and volume associated with a forced maximal exhalation by a patient, following a maximal inhalation. The reliance on a single exhalation, which most patients struggle to replicate consistently, leads to low sensitivity in early-stage disease (Johns and Crockett 2005)[3].

Besides spirometry, imaging techniques, such as computed tomography (CT) (Simon, 2005)[4], positron emission tomography (PET) (Gould et al., 2001)[5], and magnetic resonance imaging (MRI) (Tajik et al., 2002)[6], are also being used. While they provide more interpretable information, they also incur significant costs and time constraints, leading to limited use in clinical settings.

A variety of more complex Pulmonary Function Tests have started to be applied; in particular, impedance measurements and inert-gas washout. Both tests have higher sensitivity and are significantly less costly than imaging. However, increased sensitivity comes with increased complexity of output interpretation. A better-understood relationship between lung structure and test outputs is required. A way of furthering the understanding of this relationship is through the use of mathematical and computational modeling of pulmonary systems.

1.2 Aims

A fair amount of research has been developed relating to the design and computational implementation of mathematical models of airflow, blood flow, tissue mechanics, and gas transport throughout the pulmonary system. Through the use of some of these models, virtual simulations of pulmonary function tests can be performed. We can also use these simulations to probe how disease-driven changes to lung properties could affect test responses. Therefore, computational models can strengthen clinical work by helping with the interpretation of why certain physical phenomena occur.

Given these advantages, this project has been done to explore two primary aims:

- To design and implement a computational model of the human pulmonary system
- To use a Machine Learning model to interpret the pulmonary function test outputs and infer certain information about the structure of the lung that produced those outputs

Chapter 2

Biology of pulmonary systems

2.1 Anatomy

Here we will present all of the bio-mechanics of breathing that our model is going to account for.

Unless stated otherwise, the explanations in this chapter are taken from the Wikipedia page of the Respiratory System.

In humans, the anatomy of a typical respiratory system is the respiratory tract. The tract is divided into an upper and a lower respiratory tract. The upper tract includes the nose, nasal cavities, sinuses, pharynx and the part of the larynx above the vocal folds. The lower tract (Fig. 2.1) includes the lower part of the larynx, the trachea, bronchi, bronchioles and the alveoli.

The branching airways of the lower tract are often described as the respiratory tree or tracheobronchial tree (Fig. 2.1) (Gilroy, 2018)[7]. The intervals between successive branch points along the various branches of "tree" are often referred to as branching "generations", of which there are, in the adult human about 23. The earlier generations (approximately generations 0–16), consisting of the trachea and the bronchi, as well as the larger bronchioles which simply act as air conduits, bringing air to the respiratory bronchioles, alveolar ducts and alveoli (approximately generations 17–23), where gas exchange takes place (Pocock, 2006[8]; Tortora, 1987[9]). Bronchioles are defined as the small airways lacking any cartilaginous support (Gilroy, 2018)[7].

The first bronchi to branch from the trachea are the right and left main bronchi. Second, only in diameter to the trachea (1.8 cm), these bronchi (1 -1.4 cm in diameter) (Pocock, 2006)[8] enter the lungs at each hilum, where they branch into narrower secondary bronchi known as lobar bronchi, and these branch into narrower tertiary bronchi known as segmental bronchi.

The alveoli are the dead end terminals of the "tree", meaning that any air that enters them has to exit via the same route. A system such as this creates dead space, a volume of air (about 150 ml in the adult human) that fills the airways after exhalation and is breathed back into the alveoli before environmental air reaches them (Fowler, 1948)[10]. At the end

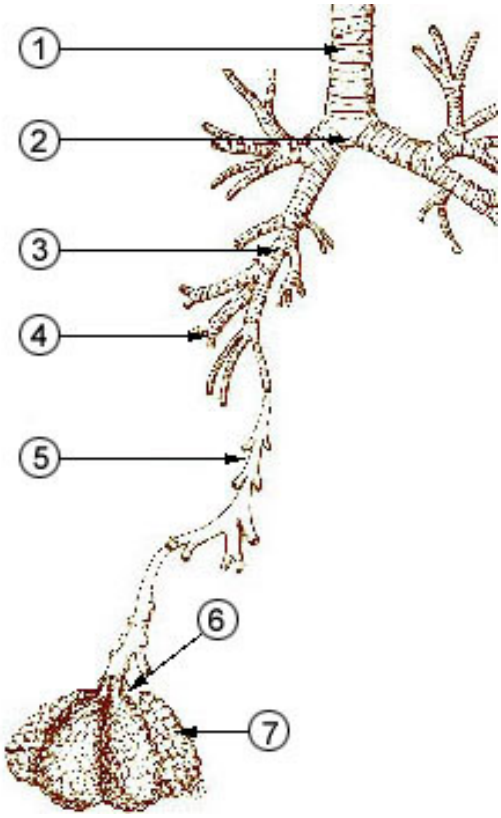


Figure 2.1:
1. Trachea
2. Mainstem bronchus
3. Lobar bronchus
4. Segmental bronchus
5. Bronchiole
6. Alveolar duct
7. Alveolus

of inhalation the airways are filled with environmental air, which is exhaled without coming in contact with the gas exchanger.

2.2 Ventilatory volumes

The lungs expand and contract during the breathing cycle, drawing air in and out of the lungs. The volume of air moved in or out of the lungs under normal resting circumstances (the resting tidal volume of about 500 ml), and volumes moved during maximally forced inhalation and maximally forced exhalation are measured in humans by spirometry. A typical adult human spirogram with the names given to the various excursions in volume the lungs can undergo is illustrated below (Fig. 2.2).

Not all the air in the lungs can be expelled during maximally forced exhalation (ERV). This is the residual volume (volume of air remaining even after a forced exhalation) of about 1.0-1.5 liters which cannot be measured by spirometry. Volumes that include the residual volume (i.e. functional residual capacity of about 2.5-3.0 liters, and total lung capacity of about 6 liters) can therefore also not be measured by spirometry. Their measurement requires special techniques (Tortora, 1987)[11].

The number of breath cycles per minute is known as the respiratory rate. An average healthy human breathes 12-16 times a minute, give an average breath length of roughly 4 seconds.

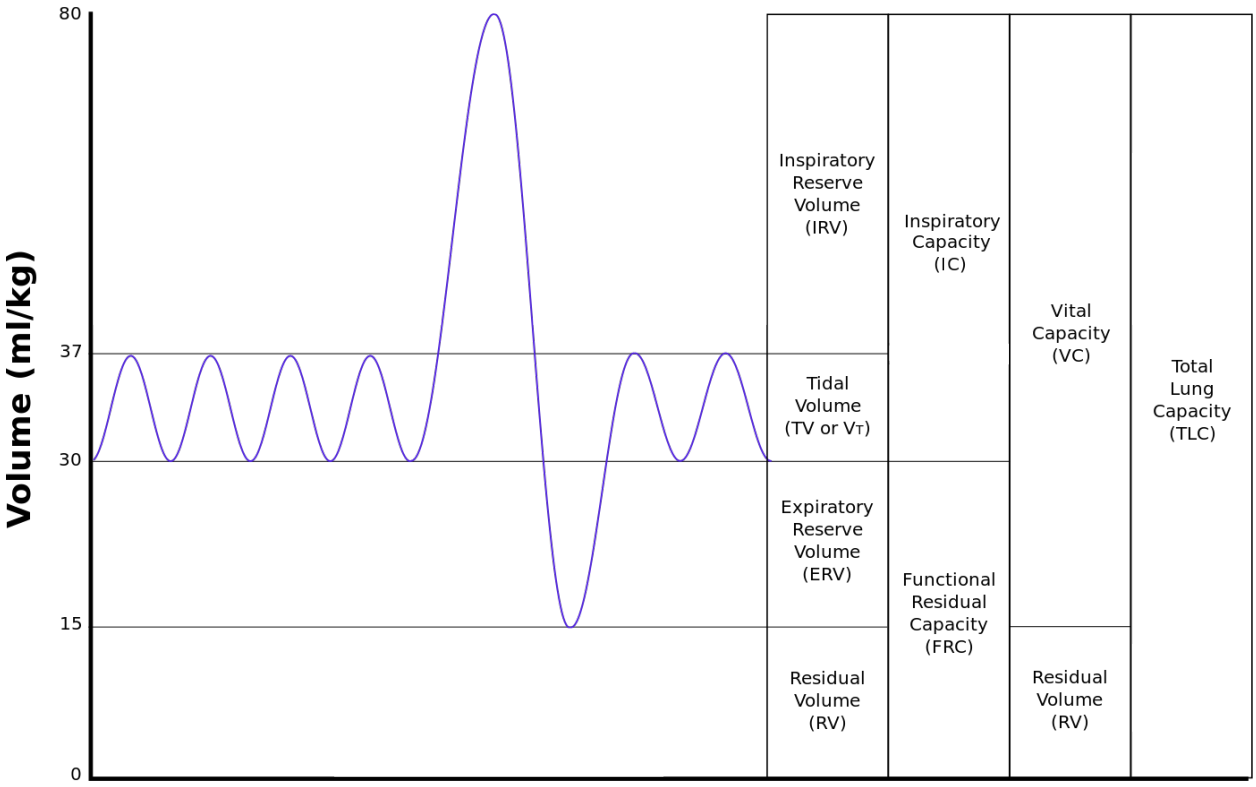


Figure 2.2: Output of a spirometer

2.3 Mechanics of Breathing

Air moves in and out of the lungs in response to differences in pressure. When the air pressure within the alveolar spaces falls below atmospheric pressure, air enters the lungs (inspiration), provided the larynx is open; when the air pressure within the alveoli exceeds atmospheric pressure, air is blown from the lungs (expiration). The flow of air is rapid or slow in proportion to the magnitude of the pressure difference. Because atmospheric pressure remains relatively constant, flow is determined by how much above or below atmospheric pressure the pressure within the lungs rises or falls.

Alveolar pressure fluctuations are caused by expansion and contraction of the lungs resulting from tensing and relaxing of the muscles of the chest and abdomen. Each small increment of expansion transiently increases the space enclosing lung air. There is, therefore, less air per unit of volume in the lungs and pressure falls. A difference in air pressure between atmosphere and lungs is created, and air flows in until equilibrium with atmospheric pressure is restored at a higher lung volume. When the muscles of inspiration relax, the volume of chest and lungs decreases, lung air becomes transiently compressed, its pressure rises above atmospheric pressure, and flow into the atmosphere results until pressure equilibrium is reached at the original lung volume. This, then, is the sequence of events during each normal respiratory cycle: lung volume change leading to pressure difference, resulting in flow of air into or out of the lung and establishment of a new lung volume. (Britannica) [12]

2.4 Lung compliance

The compliance of a system is defined as the change in volume that occurs per unit change in the pressure of the system. In layman terms, compliance is the ease with which an elastic structure stretches. Compliance is, therefore, basically a measurement of the elastic resistance of a system. Pulmonary compliance (C) is the total compliance of both lungs, measuring the extent to which the lungs will expand (change in volume of lungs) for each unit increase in the trans-pulmonary pressure (when enough time is allowed for the system to reach equilibrium) (Marshal R., 1957). To better understand pulmonary compliance, specific terminologies require a brief review. The following formula is useful to calculate compliance:

$$P_{alveolar} - P_{pleural} = \frac{1}{C} V_{alveolar}$$

Transpulmonary pressure is the pressure gradient between the inside alveolar pressure and outside pleural pressure. It mainly measures the force of lung elasticity at each point of respiration (recoil pressure). Alveolar pressure is the air pressure inside the alveoli. Pleural pressure is the pressure of the fluid present inside the space between the visceral pleura (layer adhered to lungs) and parietal pleura (chest wall lining layer). Normally the total compliance of both lungs in an adult is about 200 ml/cmH2O. Physicians rely on this concept to understand some pulmonary pathologies and help guide therapy and adjust ventilator pressure and volume settings. (NCBI, Pulmonary Compliance) [14]

Chapter 3

Flow Model

3.1 Virtual lung structure

In order to simulate processes within the lungs we need a realistic virtual model of the lungs. The structural model used here is a simplified version of the one used by the work of Brody Foy (2018)[1]. The lungs here are represented by an artificially generated binary tree structure going down to generation 16. This tree structure was implemented with the help of the NetworkX Python package. The diameter of each airway will be defined by the following equation:

$$\log d(x) = (x - N)\log(R_d S) + \log(d_n) \tag{3.1}$$

where d is the airway diameter, x is the Strahler order (Strahler, 1957)[15] of the branch, N is the maximum Strahler order of the tree, d_n is the maximum diameter, and $R_d S$ is the anti-log of the slope of the airway diameter plotted against Strahler order, taken 1.4. Equation (3.1) represents the typical rate of decrease in airway diameters, taking the tracheal diameter as initial input. This structure will be used to represent an ideal healthy patient, and will be refered to as the baseline structure.

The Strahler order of a branch is defined as the minimum number of branches between a

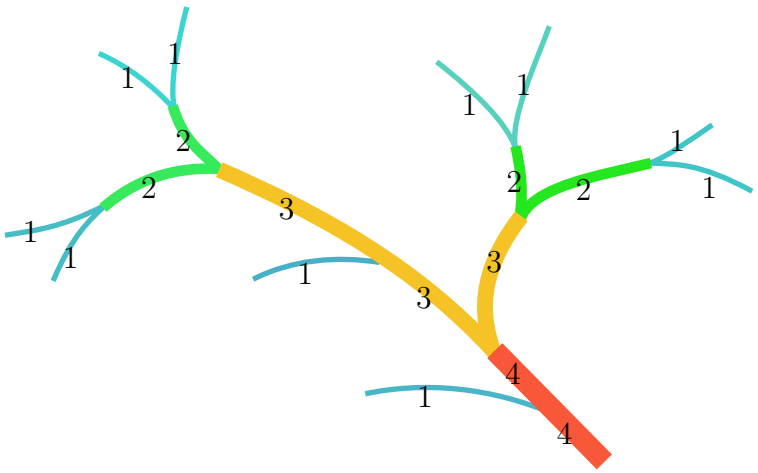


Figure 3.1: Illustration of Strahler order in a simple tree

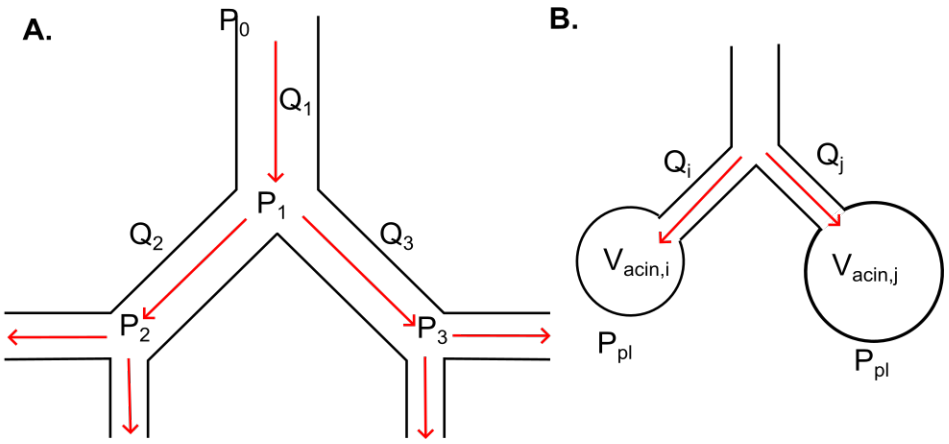


Figure 3.2: Diagram of the Ventilation model

given branch (including itself) and the nearest distal terminal branch (see Figure 3.1).

3.2 Ventilation model

We will now present a model for the simulation of various physical processes within the lungs, with the primary goal being to simulate pulmonary function tests. As a first step, the most natural model to build is one calculating ventilation and volume changes throughout the breathing cycle. We will be adopting prior models of ventilation (Mitchel et al., 2012)[16], and airway resistance (Bordas et al., 2015)[2].
To account for the respiratory zone of the lungs, each terminal branch will have a simple spherical acinar unit attached to it, with associated volume V_{acin}. We will also assume that each branch has an associated flow rate Q. Each node in the tree, representing a bifurcation point in the airways will have an associated pressure P. A simple graphic of the ventilation model is given in Figure 3.2.

Our first assumption is that flow is one dimensional and driven by the pressure difference from the pleural cavity to the trachea. Given this assumption, we can write the following equation between the pressure gradient and the flow in a branch:

$$\Delta P = R(Q)Q$$

where ΔP is the pressure difference, $Q = Au$ is the flow rate, A is the cross-sectional area of the branch, u is the airflow velocity and $R(Q)$ is the flow resistance of the branch.
For flow resistance we shall use an adaptation of Poiseuille flow adapted by Pedley et al. (1970)[17]:

$$R(Q) = \frac{32\mu Lc}{\pi d^4}\left(Re(Q)\frac{d}{L}\right)^{1/2} \tag{3.2}$$

Where d is the branch diameter and L is the branch length, μ is the viscosity of air, $c = 1.85$

is a correction constant, and $Re(Q)$ is the Reynold's number of the flow:

$$Re(Q) = \frac{4\rho|Q|}{\mu\pi d} \tag{3.3}$$

where ρ is the air density.
We will also add the condition of conservation of flow at each bifurcation:

$$Q_{parent} = \sum Q_{child}$$

where Q_{parent} is the flow of the father branch and the right hand side is the sum of the flows of the children branches.
For each terminal branch, we will assume the expansion and contraction of the associated acinar region to be driven by the flow in the branch:

$$\frac{dV_{acin,i}}{dt} = Q_i$$

and the pressure of the pleural cavity:

$$P_i - P_{pl} = \frac{1}{C}V_{acin,i}$$

where Q_i is the flow in the terminal branch, P_i is the pressure in the associated acinar region, P_{pl} is the pleural pressure surrounding the respective region, and C is the compliance of the acinar region. This model is originally adapted from the work of Ben-Tal(2006)[18].
To complete this model, we need to define two more values: we will force the pressure at the tracheal end to be equal to atmospheric pressure and the pressure at the terminal end of the tree will be a sinusoidal function:

$$P_{pl}(t) = A_{pl} + R_{pl}sin(\frac{2\pi t}{T}) \tag{3.4}$$

where A_{pl} is the mean pleural pressure, R_{pl} is the range of the pressure, and T is the breathing period, which will be set to 4 seconds.

3.3 Numerical implementation

Adding together all of the equations previously defined, we get:

$$\Delta P = R(Q)Q \qquad \text{for each branch} \tag{3.5}$$

$$Q_{parent} = \sum Q_{child} \qquad \text{for each bifurcation} \tag{3.6}$$

$$\frac{dV_{acin,i}}{dt} = Q_i \qquad \text{for each terminal branch} \tag{3.7}$$

$$P_i - P_{pl} = \frac{1}{C}V_{acin,i} \qquad \text{for each terminal branch} \tag{3.8}$$

This is a non-linear differential algebraic equation system. To implement it numerically, we will use a backward Euler discretisation (Smith, 1985) on Equation (3.7):

$$V_{acin,i}^{n+1} - dtQ_i^{n+1} = V_{acin,i}^n \tag{3.9}$$

where dt is the time step size and Q^n is the value of the flow at timestep n. Applying this discretisation to Equations (3.5)-(3.8) and rewritting them gives us:

$$\Delta P^{n+1} - R(Q^{n+1})Q^{n+1} = 0 \qquad \text{for each branch} \tag{3.10}$$

$$Q_{parent}^{n+1} - \sum Q_{child}^{n+1} = 0 \qquad \text{for each bifurcation} \tag{3.11}$$

$$V_{acin,i}^{n+1} - dtQ_i^{n+1} - V_{acin,i}^n = 0 \qquad \text{for each terminal branch} \tag{3.12}$$

$$P_i^{n+1} - \frac{1}{C}V_{acin,i}^{n+1} - P_{pl} = 0 \qquad \text{for each terminal branch} \tag{3.13}$$

In vector notation, for $\boldsymbol{x} = (\boldsymbol{P}^{n+1}, \boldsymbol{Q}^{n+1}, \boldsymbol{V}_{acin}^{n+1})^T$, this is:

$$\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{0} \tag{3.14}$$

where $\boldsymbol{P}^{n+1}$, $\boldsymbol{Q}^{n+1}$, and $\boldsymbol{V}_{acin}^{n+1}$ are vectors made up from all of the P, Q and V_{acin} variables in this order. We will solve this system using the standard Newton method (Kelley, 2003)[19], of the form:

$$\mathcal{J}\boldsymbol{dx} = -\boldsymbol{F}(x) \tag{3.15}$$

$$\boldsymbol{x} = \boldsymbol{x} + \boldsymbol{dx} \tag{3.16}$$

where $\mathcal{J}$ is the Jacobian of $\boldsymbol{F}$. In order to solve (3.14), we iteratively update $\boldsymbol{x}$ until $||\boldsymbol{F}(\boldsymbol{x})||$ falls bellow a certain tolerance tol, set as 10^{-6}. The matrix system was solved using LU factorisation (through the scipy.linalg library). This solution proved to be fast enough for the aims of this project.

One important observation is that the only values that change in the Jacobian are the ones related to Q, and so we can speed up the process by only updating those.

One thing to note is that both A_{pl} and R_{pl} must be calculated iteratively. They were initially set to -15 cmH2O and 5 cmH2O respectively, but, in order to achieve the desired FRC and breathing depth, they were adaptively chosen for each lung size. We initially compute total lung Volume (V) in a single breathing cycle and then the two values are changed by applying the following equations:

$$A_{pl} = \frac{FRC + \frac{1}{2}V_{depth}}{\frac{1}{2}(V_{min} + V_{max})}A_{pl}$$

$$R_{pl} = \frac{V_{depth}}{V_{max} - V_{min}}R_{pl}$$

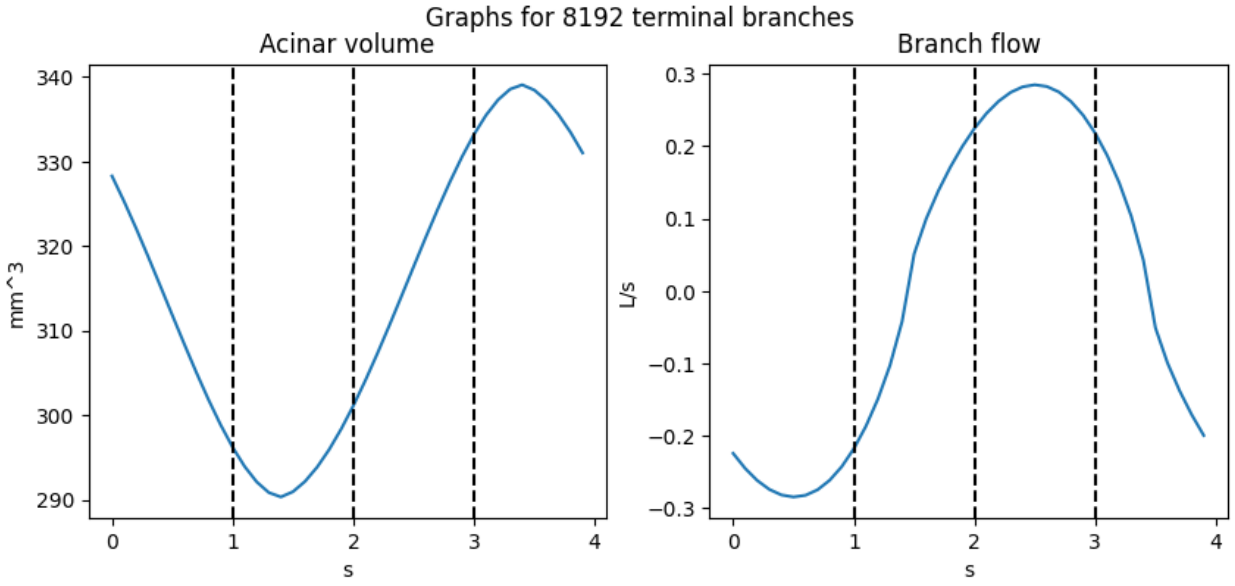


Figure 3.3: Typical ventilation model profile B

Where V_{max} and V_{min} represent the maximum and minimum values of the Pulmonary Volume computed during a breathing cycle. We keep running cycles and updating the values until the difference between the actual values and the desired values falls bellow a tolerance level *tol*.

Typical flow and volume profiles when there are 10^{13} terminal branches can be seen in Figure 3.3.

3.4 Lung domain

For the remainder of the project we will be working with lungs with 128 branches, and so 64 terminal branches. In order to create a domain space to work with, we will be decreasing the diameter of the branches. More formally, if $\boldsymbol{D}$ is the vector representing the diameters of the average, completely unrestricted lung, then $\boldsymbol{D}_{restricted}$ will be computed using this formula:

$$D_{restricted,i} = D_i * R_i \tag{3.17}$$

where $\boldsymbol{R}$ will be the vector representing the restriction profile of the lung. All other parameters will be kept fixed. Therefore, our lung domain can thought of as a subdomain of $\mathbb{R}^{128}$. However, we will have a number of restrictions on this domain:

$$R_{father} \geq R_{child} \tag{3.18}$$

$$1 \geq R_i \geq 0.1 \text{ for all i} \tag{3.19}$$

By (3.18), the children branches in the subtree of a father branch can not be less restricted than the father branch. Our model can't handle completely blocked branches and we make sure that doesn't happen through (3.19).

The generic healthy lung structure will therefore be represented by a restrictions vector filled with 1's. The values of this specific lung will then create the basis for the modelling of the whole domain. In order to speed up the simulation of a lung with a certain restrictions profile, we will initialize all of the values with that of the generic lung. We will then run breathing cycles until the volumes at the end of the breathing cycle are within *tol* of the volumes at the start. We then consider that the lung is in equilibrium.

3.5 MRI model Approximation

Now that we have a working model, we are going to create a simple test that should generally approximate an MRI scan. For each terminal branch, we will be assessing its maximum volume during a normal breathing cycle. If $V^n_{acin,i}$ is the volume of the acinar region i at time step n, then we will have the following equation:

$$V_{max,i} = \max(V^n_{acin,i}) \quad \text{for all time steps } n \text{ in a breathing cycle} \tag{3.20}$$

However, we know that the only variability in the domain of our lungs is strictly given by the restrictions vector. For our use case we can then define a function $Test$ that takes as input a restrictions vector and gives as output a vector of maximum acinar volumes:

$$\boldsymbol{Test}(\boldsymbol{R}) = \boldsymbol{V_{Max}} \tag{3.21}$$

Two test outputs of a largely unrestricted lung (blue) and very restricted lung (orange) respectively can be seen in Figure 3.4.

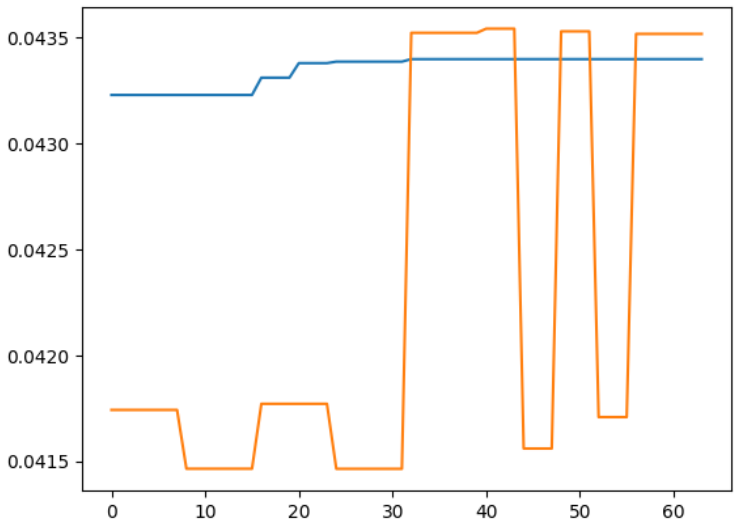


Figure 3.4: Test outputs of two restrictions profiles

The volumes are represented in two dimensions because our model only has two dimensions. This is a crude estimation of an MRI scan which would be represented in three dimensions using voxels with colours on a gradient representing how well ventilated each area is.

Chapter 4

Machine Learning Model

The task of our ML model is to infer the lungs structure that might have given a certain set of results to the MRI scan we devised. We will intuitively represent it using a function ***Diagnose***:

$$\boldsymbol{Diagnose} \simeq \boldsymbol{Test}^{-1} \tag{4.1}$$

$$\boldsymbol{Test}(\boldsymbol{R}) = \boldsymbol{V_{Max}} \tag{4.2}$$

$$\boldsymbol{Diagnose}(\boldsymbol{V_{Max}}) \simeq \boldsymbol{R} \tag{4.3}$$

This is what a clinician would like to do when looking at an MRI scan.

We are therefore going to be predicting numeric values; All of the information is potentially important and so we don't require dimension reduction. This is also a supervised model and we're most looking for accuracy. All of these considerations leave us with three options: Random Forests, Neural Networks and Gradient Boosting Trees.

We have decided that Neural Networks would be the best choice.

4.1 Neural Network Model

For this task, a fully connected neural network with 4 hidden layers of sizes [500, 100, 100, 100], with a SELU activation function after each one was chosen.

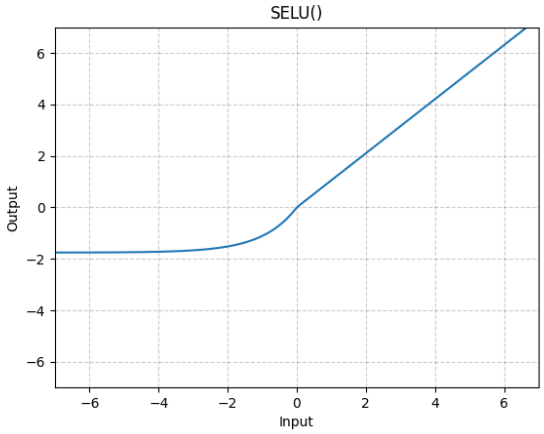
SELU (Scaled Exponential Linear Unit) is an activation function used in artificial neural networks. It aims to address the vanishing and exploding gradient problem in deep neural networks. The SELU activation function is defined as:

$$\text{SELU}(x) = \text{scale} * (\max(0, x) + \min(0, \alpha * (\exp(x) - 1))) \tag{4.4}$$

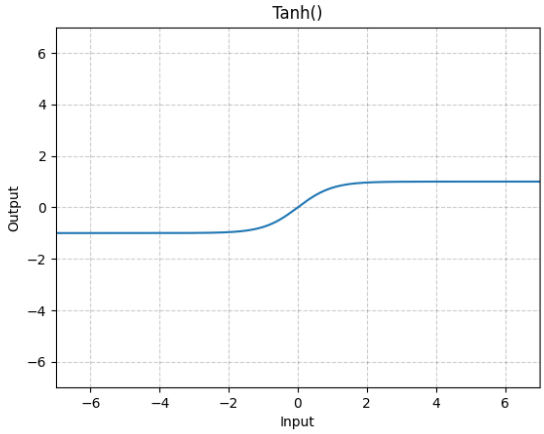
where α is ~ 1.67 and scale is ~ 1.05.

At the end, a Tanh layer along with the appropriate scaling was added. The Tanh function equation looks like this:

$$\text{Tanh}(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)} \tag{4.5}$$



SELU function graph



Tanh function graph

From both the equation and the figure graph we can see that Tanh always outputs values in the range [-1, 1]. This is very useful to us because we know that our outputs (the restrictions) are values in the range [0.1, 1]. With the correct scaling at the end, we can therefore make sure that our predictions will always be in the appropriate domain.

We used Adam, one of the best algorithms for stochastic gradient descent. This was implement in python with Pytorch and Pytorch Ignite, a high-level library that helps with the training and evaluation of neural networks. This seemed to be the best configuration given the task.

Smaller networks would not be strong enough to learn the underlying structure and so would underfit, while larger networks would be very prone to reaching inescapable local minimas. We have also experimented with the Batch Normalisation and Instance Normalisation provided by Pytorch but they're performance is rather limited. These would keep a running mean and standard deviation for the features which would be updated after each batch. We have found that simply pre-computing the means and standard deviations of the features and loading them onto the GPU before the computation gave the best results.

We have set up a linear learning rate scheduler that would decrease the learning rate hyperparameter after each iteration. Generally, a starting learning rate of 0.001 worked best.

4.2 Computational Constraints

All of the following computations were done on my personal computer which contains:

- Intel(R) Core(TM) i5-10210U @ 1.60GHz CPU
- Nvidia GeForce MX330 2GB GPU
- 8GB RAM

This CPU specifically has 4 cores and 2 threads per core, so a maximum total of 8 concurrent processes. The computation of separate data points is parallelizable and so we decided to

use the CPU for the creation of datasets. For this task we used Multiprocessing library in Python. As the graph from Figure 4.1 below shows, it turns out that 8 processes is indeed the most efficient number of concurrent processes to have. Adding more processes after that point brings no improvement and might even slow down the computation due to context-switching.

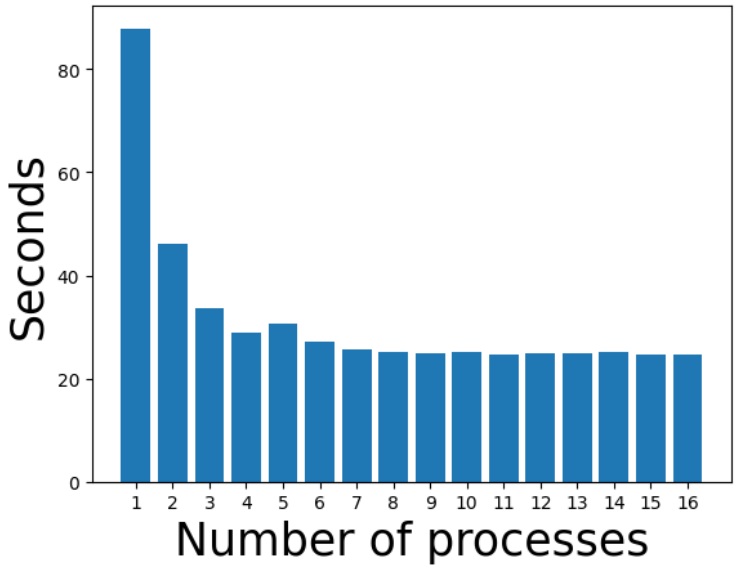


Figure 4.1: Total CPU time usage depending on the number of concurrent processes; in each case, a total of 1000 entries were computed.

We will however use the GPU for the training of the neural network because it is much faster at the required type of computation. Since it only has 2GB of VRAM and our full datasets greatly exceed that size. To get around this limitation, we will store the data using the NPY format provided by numpy.lib.format because we can then use numpy.memmap to access these values without loading the entire dataset into VRAM. However, it still needs to be loaded into RAM. Memmap is a memory map format provided by Numpy.

4.3 Dataset

We will now build a dataset for the model to learn on. In order to build the restrictions list, we first choose a the number of areas N that will be constricted by some value. Every other branch will be completely healthy. We then create a vector $\boldsymbol{DB}$ of N randomly chosen distinct branches. The restrictions will be sequentially computed in the following manner:

$$\boldsymbol{R}_{child} = \begin{cases} \max(\boldsymbol{R}_{father} * \boldsymbol{X}, 0.01) & \text{if } father \text{ is inside } \boldsymbol{DB} \\ \boldsymbol{R}_{father} & \text{if } father \text{ is not inside } \boldsymbol{DB} \end{cases}$$

where $\boldsymbol{X}$ is a random variable such that:

$$\boldsymbol{X} = Normal(\frac{low_{case} + high_{case}}{2}, 0.02) \tag{4.6}$$

where $Normal(\mu, \sigma)$ represents a normal random variable with mean μ and standard deviation σ, and *case* represent the case number associated with the current dataset.

We have decided to break it up into 16 distinct datasets that will have two varying properties: Dn and [*low*, *high*]. Dn will represent the number of diseased nodes and [*low*, *high*] will represent the range of values by which the respective nodes will be restricted. Dn will come from the set {2, 3, 4, 5} and [*low*, *high*] will come from the set {[1, 0.85], [0.85, 0.7], [0.7, 0.55], [0.55, 0.4]}. The graph has a depth of 7, but we will restrict the set of diseased nodes to a maximum of depth 5 (therefore to the first 32 branches of the tree).

For this dataset, the computation times and memory requirements for the creation of the data points are shown in the table below:

Dataset Size (Data points)	Computation Time	Memory Size
1	25 ms	1.5 KB
10^3	25 s	1.5 MB
$3 * 10^5$	2.08 hrs	450 MB
10^6	6.94 hrs	1.5 GB
$16 * 10^6$	4.62 days	24 GB

Since we only disease the first 32 branches, we will also only ask the neural network to predict the restrictions on these 32 branches. The rest of the restrictions will be automatically filled with the restriction of the father branch. To assess how well the model learns, we have two graph plots for each case. The first one (Figure 4.2) plots $\boldsymbol{D}$:

$$\boldsymbol{Test}(\boldsymbol{R}) = \boldsymbol{V_{Max}} \tag{4.7}$$

$$\boldsymbol{Diagnose}(\boldsymbol{V_{Max}}) = \hat{\boldsymbol{R}} \tag{4.8}$$

$$\boldsymbol{D} = \max(\text{absolute}(\boldsymbol{R} - \hat{\boldsymbol{R}})) \tag{4.9}$$

So $\boldsymbol{D}$ is the vector representing the maximum absolute difference of any two corresponding real and predicted restrictions. This set of graphs is a good representation of how well the model happens to predict the particular solution give by the each datapoint. We believe that the lung model is complex and that $\boldsymbol{Test}$ is not a 1-1 function. It is very likely that there is a large domain of restrictions vectors that return very similar values for the volumes vectors. In order to assess how well our model predicts this underlying structure, we will look at Figure 4.3 which plots $\boldsymbol{Dr}$:

$$\boldsymbol{Test}(\hat{\boldsymbol{R}}) = \hat{\boldsymbol{V}}_{\boldsymbol{Max}} \tag{4.10}$$

$$\boldsymbol{\mu} = Mean(\boldsymbol{V_{Max}}) \tag{4.11}$$

$$\boldsymbol{\sigma} = Variance(\boldsymbol{V_{Max}}) \tag{4.12}$$

$$\boldsymbol{s}\hat{\boldsymbol{V}}_{\boldsymbol{Max}} = \frac{\hat{\boldsymbol{V}}_{\boldsymbol{Max}} - \boldsymbol{\mu}}{\boldsymbol{\sigma}} \tag{4.13}$$

$$\boldsymbol{sV_{Max}} = \frac{\boldsymbol{V_{Max}} - \boldsymbol{\mu}}{\boldsymbol{\sigma}} \tag{4.14}$$

$$\boldsymbol{Dr} = \max(\text{absolute}(\boldsymbol{s}\hat{\boldsymbol{V}}_{\boldsymbol{Max}} - \boldsymbol{sV_{Max}})) \tag{4.15}$$

The idea is then that we run the diagnosed restrictions back through the MRI scan test and compare the recomputed volumes with the original ones. Since the volumes have really small values with even smaller changes depending on the restrictions, we will scale them for the purpose of a much better human visualisation of the results. This second table of graphs is the one that represents the overall results and quality of the resulting neural network.

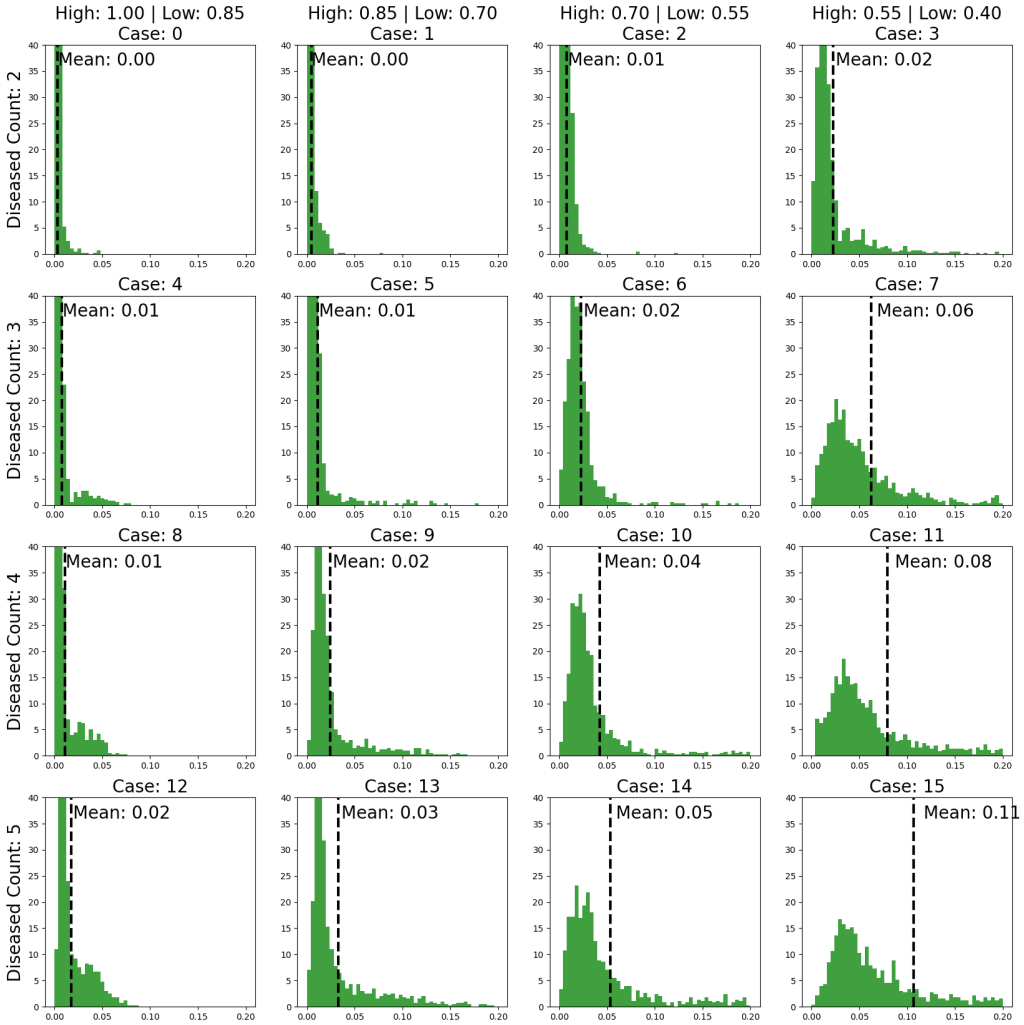


Figure 4.2: Maximum difference between Real and Predicted restrictions

We generally expect to observe a bell curve in most of the graphs mentioned above. We also expect the bell curve to move to the right and flatten as the restrictions space gets more and more complex. This is because the underlying structure should be become harder to

learn. This is indeed the observed result: if we only have two diseased branches (the first row of the graphs), or only have light restrictions (the leftmost column of the graphs), the diagnosis is fairly accurate: we are on average off by a maximum of 0.1 standard deviations.

However, each extra diseased nodes raises the number of total possible combinations of diseased nodes by at least one order of magnitude. At the same time, while lightly restricted branches might have strong consequences only in their local region, heaviliy restricted branches are very likely to affect the ventilation pattern of the whole lung. Therefore, the neural networks has to pick up on correlations that are both more numerous and more complex.

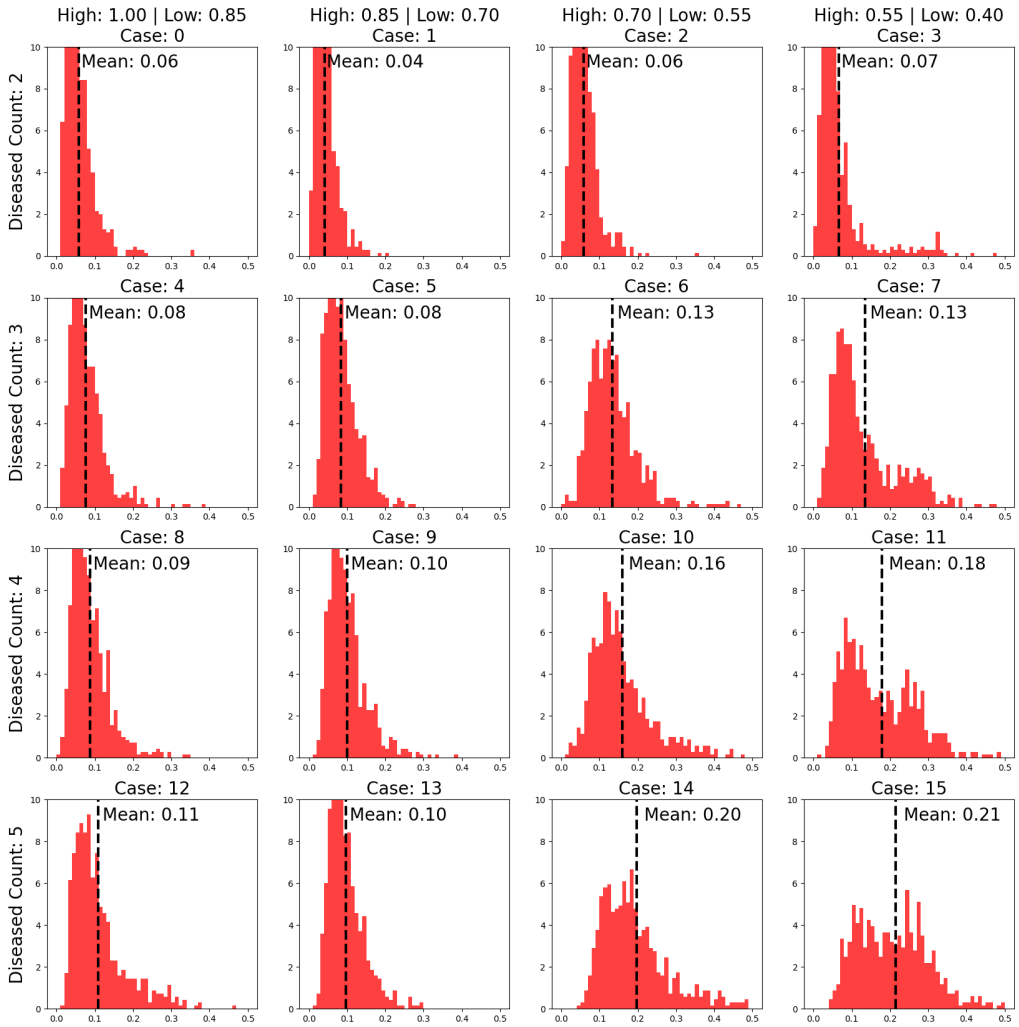


Figure 4.3: Maximum difference between Real and Recomputed Scaled Volumes

Chapter 5

Conclusion

To recapitulate: We have first adopted a mathematical model of the human lungs, presented mainly by the work Brody Foy. We then implemented this model in Python using multiple libraries and packages (linalg, NetworkX, scipy etc.) and used it to devise a lung function test mimicking real MRI scans. Given this test, we built and trained a neural network capable of diagnosing the test results fairly accurately for a structure made up of 128 branches and between 2 and 5 diseased branches.

However, the time and memory complexity of the task is exponential in the depth of the model. For 1024 nodes, we would already be reaching a hundred Gigabytes of memory, making larger tasks unfeasible. Still, there are possible solutions: the dataset used for training the ML model doesn't have to be kept in memory; new data points could be computed for every iteration, trained on, and then discarded.

One other issue is the applicability of any resulting Machine Learning models: no two human lungs are truly the same and so any pre-trained model would struggle when faced with a real-world test result. One solution could be in the choice of representation for test results and the ML model: if the results of the MRI scan were to be kept in a fixed size and scale image format, all lungs could be represented, even with slightly varying sizes. Something like a convolutional neural network, which would also take advantage of the spatial regularities, could then be used.

Bibliography

[1] Foy, B. (2018). Applied mathematical modelling of pulmonary function tests [PhD thesis]. University of Oxford.

[2] Bordas R, Lefevre C, Veeckmans B, Pitt-Francis J, Fetita C, Brightling CE, Kay D, Siddiqui S, Burrowes KS. Development and Analysis of Patient-Based Complete Conducting Airways Models. PLoS One. 2015 Dec 11;10(12):e0144105. doi: 10.1371/journal.pone.0144105. PMID: 26656288; PMCID: PMC4684353.

[3] Johns, D. P. and Crockett, A. J. (2005). Lung function testing. Evidence-based Respiratory Medicine, page 25.

[4] Simon, B. A. (2005). Regional ventilation and lung mechanics using X-Ray CT. Academic Radiology, 12(11):1414–1422.

[5] Gould, M. K., Maclean, C. C., Kuschner, W. G., Rydzak, C. E., and Owens, D. K. (2001). Accuracy of positron emission tomography for diagnosis of pulmonary nodules and mass lesions: a meta-analysis. Journal of the American Medical Association, 285(7):914–924.

[6] Tajik, J. K., Chon, D., Won, C., Tran, B. Q., and Hoffman, E. A. (2002). Subsecond multisection CT of regional pulmonary ventilation. Academic Radiology, 9(2):130–146.

[7] Gilroy, A.M., MacPherson, B.R., Schulte, E., Schuenke, M., Ross, L.M.(2008). Atlas of Anatomy.Germany: Thieme, pages 108–111.

[8] Pocock, Gillian; Richards, Christopher D. (2006). Human physiology: the basis of medicine (3rd ed.). Oxford: Oxford University Press. pp. 315–317.

[9] Tortora, Gerard J.; Anagnostakos, Nicholas P. (1987). Principles of anatomy and physiology (Fifth ed.). New York: Harper Row, Publishers. pp. 556–586

[10] Fowler W.S. (1948). "Lung Function studies. II. The respiratory dead space". Am. J. Physiol. 154 (3): 405–416.

[11] Tortora, Gerard J.; Anagnostakos, Nicholas P. (1987). Principles of anatomy and physiology (Fifth ed.). New York: Harper Row, Publishers. pp. 570–572

[12] https://www.britannica.com/science/human-respiratory-system/The-mechanics-of-breathing

[13] MARSHALL R. The physical properties of the lungs in relation to the subdivisions of lung volume. Clin Sci. 1957 Aug;16(3):507-15. PMID: 13473164.

[14] https://www.ncbi.nlm.nih.gov/books/NBK538324/

[15] Strahler, A. N. (1957). Quantitative analysis of watershed geomorphology. Transactions American Geophysical Union, 38(6):913–920.

[16] Mitchell, J. H., Hoffman, E. A., and Tawhai, M. H. (2012). Relating indices of inert gas washout to localised bronchoconstriction. Respiratory Physiology & Neurobiology, 183(3):224–233.

[17] Pedley, T., Schroter, R., and Sudlow, M. (1970). The prediction of pressure drop and variation of resistance within the human bronchial airways. Respiration Physiology, 9(3):387–405.

[18] Ben-Tal, A. (2006). Simplified models for gas exchange in the human lungs. Journal of Theoretical Biology, 238(2):474–495.

[19] Kelley, C. T. (2003). Solving nonlinear equations with Newton's method, volume 1. Siam.