APL405 PROJECT GROUP3

Group 3:
Shresth Somya
Yuvraj Verma
Vishal Kumar Saini
Rahul Saini

15 March 2022

Contents

1 Introduction

1.1 Purpose of Stress analysis of Aorta

There are possible malfunctions in Aorta, which includes rupturing and bulging. Ascending aortic aneurysm (AsAA) rupture is believed to occur when the mechanical stress acting on the wall exceeds the strength of the wall tissue. Therefore, knowledge of the stress distribution in an intact AsAA wall could be useful in assessing its risk of rupture.In this study, by using machine learning techniques, a deep learning (DL) model is developed to directly estimate the stress distributions of the aorta.

1.2 Existing FEA model vs ML-FE model

Structural finite-element analysis (FEA) has been widely used to study the biomechanics of human tissues and organs, as well as tissue–medical device interactions, and treatment strategies. However, patient-specific FE models usually require complex procedures to set up and long computing times to obtain final simulation results, preventing prompt feedback to clinicians in time-sensitive clinical applications. In this study, by using machine learning techniques, we developed a deep learning (DL) model to directly estimate the stress distributions of the aorta.

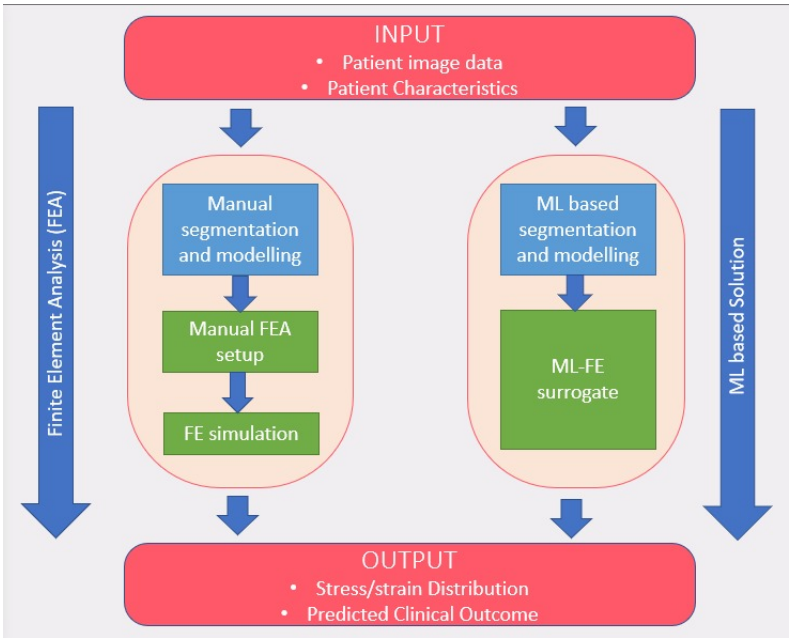

Figure 1: FEA vs ML model

The current implementation of FEA model is shown in Figure 1:(i) patient anatomic geometries are obtained mostly through manual annotation of in vivo clinical image data (e.g. image region segmentation and object boundary delineation); (ii) a Finite Element model is set up by specifying material properties and the boundary and loading conditions in the Finite Element model; (iii) the Finite Element model is submitted as an FEA job, to a finite-element numerical solver to obtain simulation results. Depending on the complexity of the model and application, it may take anywhere from minutes to days. Therefore, impractical to apply patient-specific computational analyses in time-sensitive clinical applications.

To resolve these limitations, a machine learning (ML) approach is taken (shown in Figure 1): (i) an ML based geometry reconstruction and modelling is used to directly output patient anatomical geometries from input patient raw clinical image data; (ii) deep learning (DL) techniques are used to build deep neural networks (DNNs) as an FEA surrogate for mechanical stress analysis.

On implementing both the approach with the given input shape of thoracic aorta geometry, the trained deep learning (DL) model can output the aortic wall stress distributions within 1 s, while FEA takes about half an hour on on the same computer.

2 Problem Statement

The Frameworks is shown in Figure 2:(i) The shape data is encoded to represent it by small number of scalar values; (ii) A non-Linear mapping is performed with deep learning to find the stress code from the encoded shape; (iii) Stress code obtained is decoded into wall stress distributions.

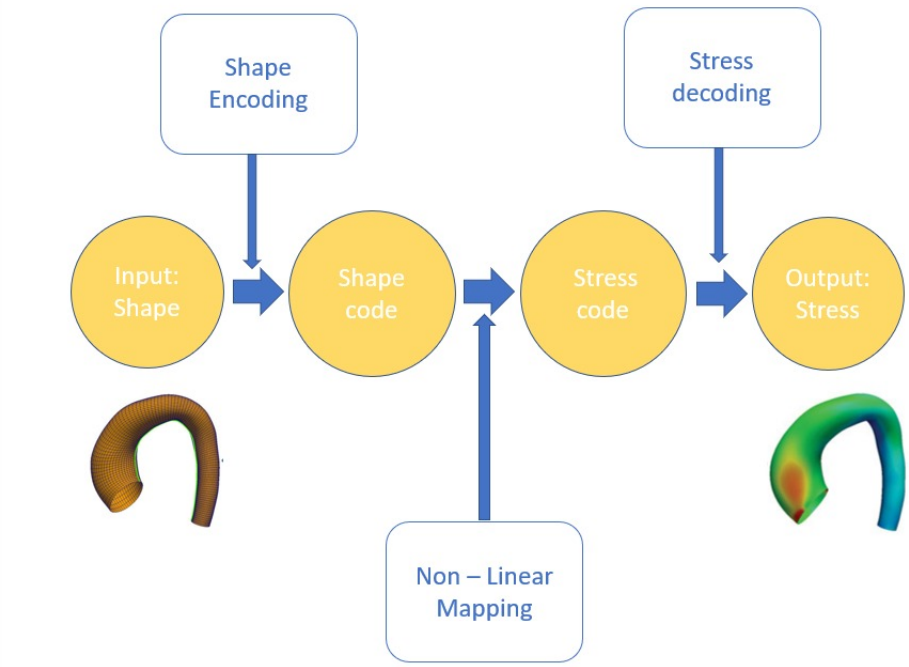

Figure 2: Workflow

2.1 Shape Encoding

Shape encoding refers to a method of representing a shape by a small number of scalar values. Principle component analysis (PCA-based shape analysis) is performed, which is equivalent to a neural network with linear units and without hidden layers. It is shown in Figure 3. The shape data has 5000 nodes with 3 coordinates, and a total of 729 shapes. Thus the size of shape data is (15000 x 729). We are encoding it with the equation 1.

$$\alpha_m = \frac{W_m^T(X - \overline{X})}{\sqrt{\lambda_m}} \tag{1}$$

In equation 1, W is the eigen vector of shape variation , λ is its eigen values and $\overline{X}$ is mean shape. The link, i.e., weights in the figure for respective sigma shape code is the respective component of PCA.

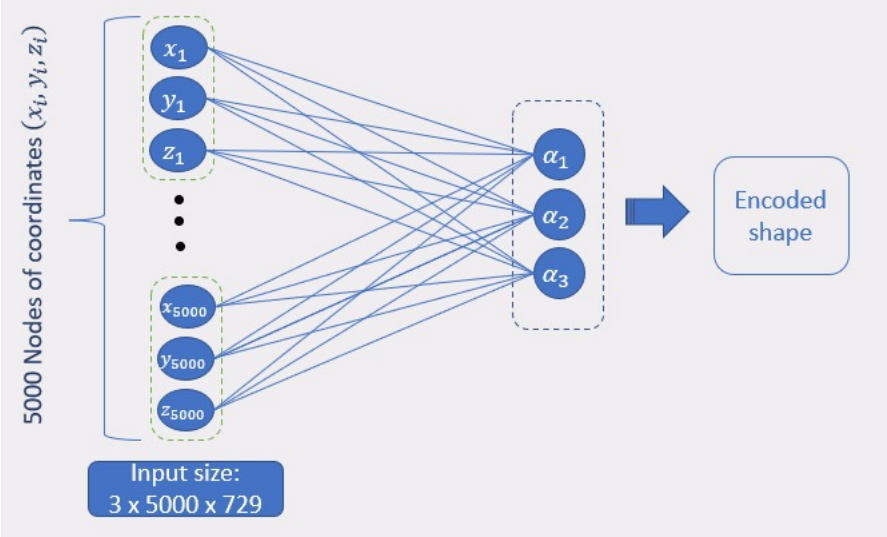

Figure 3: Neural Network for shape encoding

2.2 Non-Linear Mapping

The encoded shape data is mapped to stress data through, nonlinear mapping with two hidden layers each having 128 neurons as shown in Figure 4. Non-linear regression is applied with soft-plus units. ReLU or other activation function units can be also used for the same purpose.

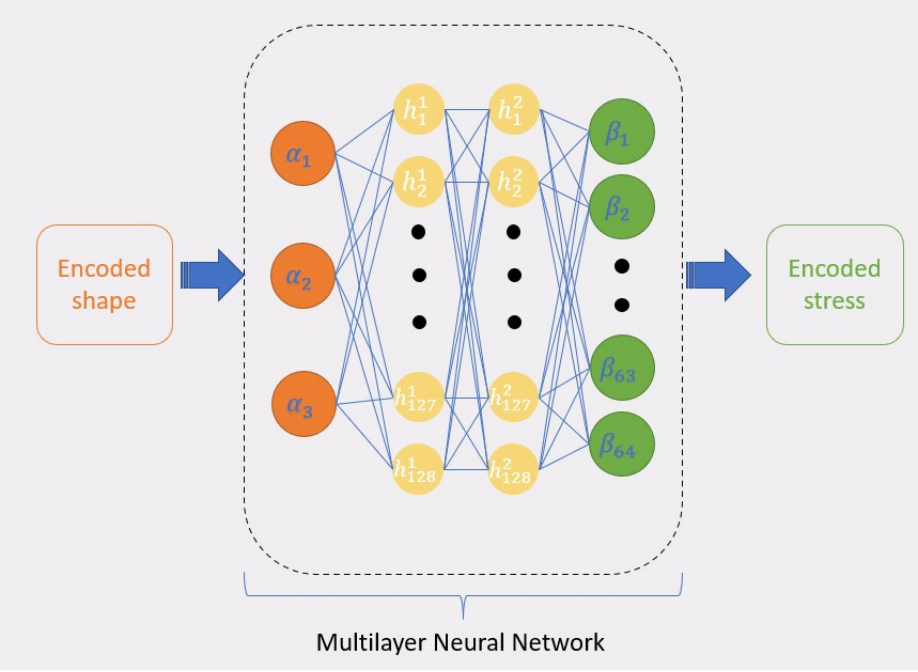

Figure 4: Non-Linear Mapping

2.3 Stress encoding and decoding

For decoding and encoding of stress is done by bi-directional neural network with multiple hidden layers as shown in Figure 5. Stress encoding is done only once for training stage while decoding is the part of the DL Model.To facilitate the processes, each aorta mesh was divided into 25 regions with 200 nondes in each region. Low rank approximation (LRA) was used for stress encoding and decoding.

The wall stress distribution is represented by S, where each column vector $S^{(k)}$ has 600 stress values per region. The three stress components are S_{11} along the circumferential direction, S_{22} along the longitudinal direction and the shear stress S_{12}. In figure 5, it is represented by light grey coloured circle an is the last layer from left to right. The middle layer is represented by R, where $R = [R^T_{region\_1}, R^T_{region\_2}, ..., R^T_{region\_25}]$ and $R_{region\_k} = [r_1^{(k)}, r_2^{(k)}, ..., r_{256}^{(k)}]$, which is the stress code for the k^{th} region.

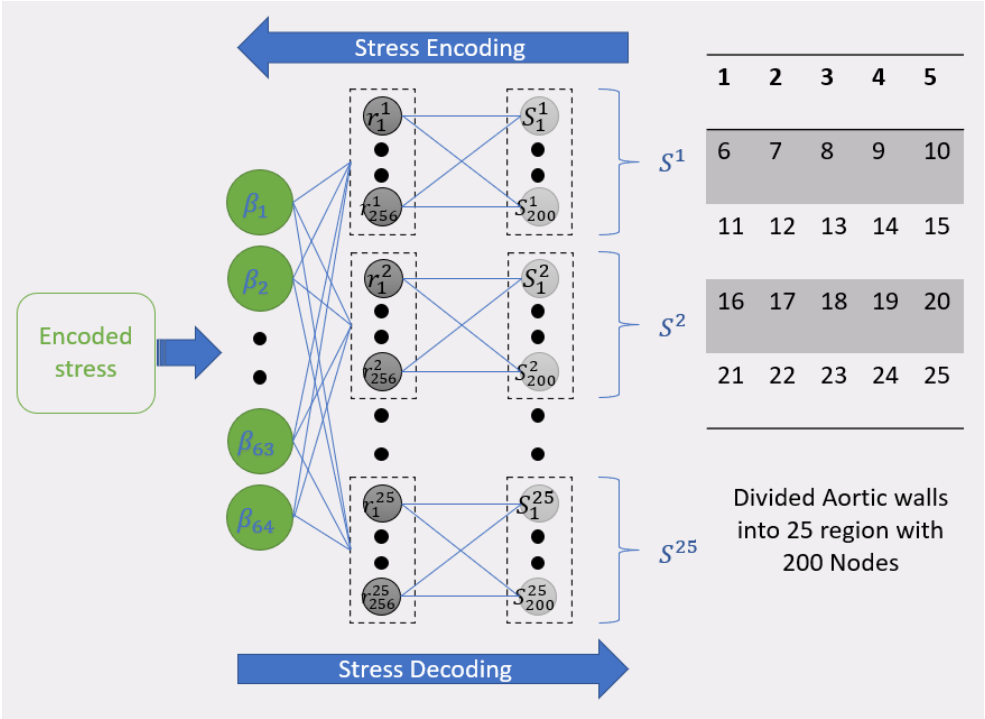

Figure 5: Stress Encoding and Decoding using Bidirectional NN

2.3.1 Stress Decoding

For decoding of stress code we have to move from left to right in the Figure 5. We are using transposed convolution layers for neural network. Using the

Equation 2, given below the middle layer is calculated.

$$R = \sum_{n=1}^{64} \beta_n \mu_n U_n \tag{2}$$

LAR was performed on R, U_n are the singular vectors, μ_n are respective singular vector and β_n are the LRA parameters. The 64 most significant singular components were retained in the LRA, which led to an approximation error less than 1%.
Using the Equation 3, given below wall stress is calculated.

$$S^{(k)} = \sum_{i=1}^{256} r_i^{(k)} v_i^{(k)} V_i^{(k)} \tag{3}$$

From LRA performed on $S^{(k)}$, $V_i^{(k)}$ are the left-singular vectors, $v_i^{(k)}$ are the corresponding singular values and $r_i^{(k)}$ are the LRA parameters. The 256 most significant singular components were retained in the LRA, which led to an approximation error less than 1%.

2.3.2 Stress Encoding

For encoding of wall stress we have to move from right to left in the Figure 5. We are using convolution layers for neural network.

$$r_i^{(k)} = \frac{(V_i^{(k)})^T S^{(k)}}{v_i^{(k)}} \tag{4}$$

$$\beta_n = \frac{U_n)^T R}{\mu_n} \tag{5}$$

The above equations 4 and 5 are derived from the Equations 2 and 3 to perform the encoding of wall stress distribution. Encoding of stress data is done only once.

2.3.3 Assumption for Better computation

Two assumption where taken:

- All the 25 regions shares the same weight for the neural network.
- The same weights are shared for both encoding and decoding.

3 Future Works

3.1 Reason for encoding and decoding

The reason behind encoding and decoding of stress and shape data is not explicitly mentioned in the research paper. But we have a hypothesis that it is used only to reduce the dimension of data. Looking at the shape and stress data, many components are zero, therefore have many reluctant information which can be compressed.

For shape data, reducing the size from (15000,729) to (3,729) stores approximately 80.1% variance. This reduces the computation cost. Similarly, the stress data is encoded to size (64,729) which has less than 1% approximation error.

3.2 Plans for future

We didn't have the facility to execute our codes therefore cannot conclude any results. We are planing to run the code with encoding and decoding and without encoding and decoding and compare the results and time taken. Framework of code without encoding and decoding is mentioned in Figure 6.

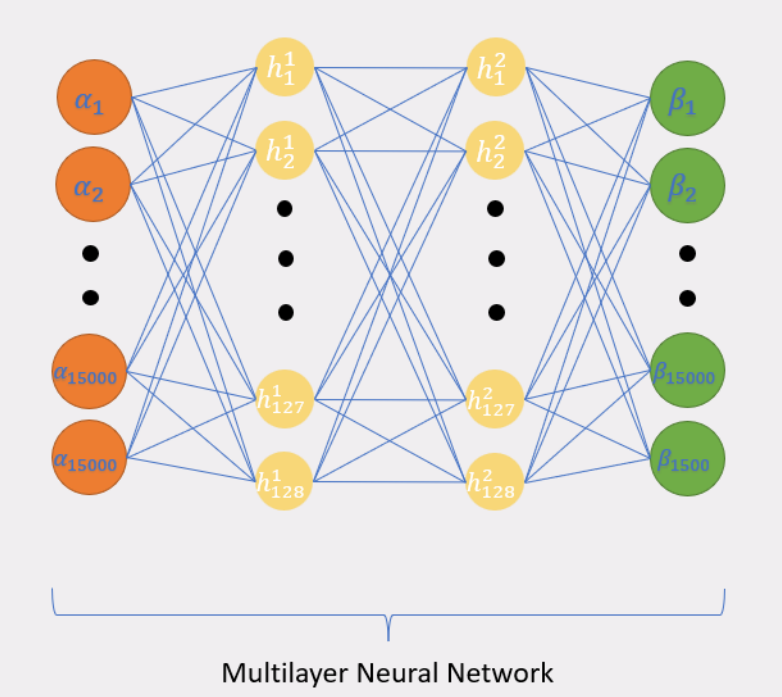

Figure 6: Deep Neural Network without encoding and decoding of stress or shape data