Integrated Neural Network and Finite Element Analysis for constitutive modelling of soil

Keshav Kashichenula

In partial fulfilment of the requirements for the degree of

Master of Sciences

In Geo-Engineering

at Delft University of Technology

Thesis Committee

Dr. P.J. Vardon,	TU Delft
Dr. ir. C. Zwanenburg,	TU Delft
Dr. R. Taormina,	TU Delft
Dr. J. Nuttall,	Deltares
E. Smyrniou,	Deltares

TU Delft Delft University of Technology

Deltares Enabling Delta Life

Acknowledgements

This thesis is an attempt to identify a potential bridge between Machine Learning and Geotechnical engineering. Working on this thesis, I was able to develop my skills further in the area of Geotechnical engineering and in programming. Ultimately, I am satisfied with the achieved results and the knowledge I gained from it. It would not have been possible to deliver this without any help.

I would like to thank Dr. Phil Vardon and Dr. Jonathan Nuttall and Eleni Smyrniou from Deltares for providing me such an interesting project. I would like to thank Dr. Phil Vardon and Dr. Jonathan Nuttall for their guidance in necessary situations and sharing their insights at various points in the thesis which helped me in the betterment of the thesis. I would also like to thank Eleni Smyrniou who guided me on a day to day basis with her knowledge in neural networks and FORTRAN. Her insights on various solutions and her experience in this field, enabled me to look at the problem through various perspectives. I would like to extend my gratitude towards Dr. Riccardo Taormina for helping me through the difficult parts in the neural networks and guiding me in the right direction in terms of developing my skills in machine learning. I would like to thank Dr. Cor Zwanenburg for providing me with his insightful remarks at various points in my thesis in the aspect of Geotechnical engineering which helped me in understanding the concepts and looking at the problem from a practical point of view.

I would also like to extend my gratitude to Dr. Ronald Brinkgreve for helping in obtaining a license for the PLAXIS software without which it would not have been possible to move ahead.

Also, I would like to thank my parents and friends who stood by my side throughout my student life and especially for being a moral support during the tough times of this pandemic.

Keshav Kashichenula, 2021

Abstract

Finite Element Analysis is an application used in many engineering fields, including geotechnical engineering. With increasing complexities of structures and material behavior, classical approach for non-linear finite element analysis has become computationally expensive(Gulikers, 2018). In order to improve the efficiency and accuracy of the model, Machine Learning (ML) is introduced. The ever-increasing accessibility and magnitude of computational power has made machine learning gained popularity since 1990s. Artificial Neural Networks have especially been proven to be a versatile machine learning framework, capable of solving complex tasks such as voice recognition to solving failure analysis of structures.

This thesis aims at implementing soil constitutive models such as Linear Elastic and Elasto-plastic models and integrating them into a Finite Element Engine. Two different approaches have been attempted to implement the ANN into the constitutive models.

The first approach involves the implementation of a "Generic" Neural network to simulate the natural anisotropic behavior of soils. Datasets have been generated with the help of theoretical soil tests classified as 'Standard' and 'Non-standard' tests. The standard tests depict an isotropic behavior, where the strains and stresses in different directions are proportionate. Non-standard tests consist of data representing anisotropic behavior where the strain increments in every direction are randomized. The neural network has first been implemented for a linear elastic soil model. It is observed that when the network is trained and tested with similar kind of data, predictions are accurate. However, when the neural network is trained with a certain type of data(such as standard test), it is unable to predict the response for the other data(non-standard test). When the neural network (trained with standard data) is subsequently implemented in Finite Element analysis by deriving and substituting the material stiffness matrix from the neural network in a constitutive model, the difference between the isotropic linear elastic matrix (for linear elastic model) and the matrix derived from the neural network lead to significant errors, despite the accurate predictions of the neural network.

The second approach is a component based approach in which the neural network is modeled to learn the typical behaviors of soil such as linear elasticity and plasticity. The data for training and testing the neural network are generated using standardized laboratory tests (CRS, DSS, Biaxial etc). The neural network is modeled for linear elastic soil, by applying certain weight constraints. It is observed that the neural network has accurate predictions. Further implementation in constitutive model shows that the neural network is able to mimic the behavior of Linear elastic soil model. The neural network model, is subsequently implemented to predict Mohr-Coulomb behavior, and the predictions are significantly accurate. Further the generalization ability of the network is analyzed to evaluate the practical applicability of the network, followed by a sensitivity analysis to get an understanding about the influencing parameters in the network.

The devised neural network, when implemented in a constitutive model displays a behavior similar to the theoretical constitutive model of Mohr-Coulomb soil. Finally, this thesis concludes with the implementation of the neural network constitutive model in Finite Element Analysis where it leads to convergence errors in the model.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In civil engineering, materials such as steel, concrete, soil etc., are continuously studied for their properties and response to various environmental conditions in order to improve the structural designs. Each material that is used in the design calculations is considered as a continuous medium, and the physical and mechanical principles such as the law of conservation of energy are used for analysis of the structure(Popa and Batali, 2010). A constitutive model is thus used, which describes the material response to various external actions.

A constitutive model is prepared according to the material properties of the material under test. The model could vary from a simple linear relation (between stresses and strains) to an extremely complex equation, based upon the said material properties. The constitutive model is a mathematical equation devised to explain the behavior or response of the material under various conditions. With the help of a Finite Element Method (FEM), more and more constitutive models can be used to explain the complex behavior of materials. Since its introduction in the 20th century, FEM has gained immense popularity in the field of engineering as a tool for analyses. FEM, along with the constitutive equations has been an aide in developing a relation between stresses and strains in various solids to study their behavior/response. This method has since been extended to the field of Geotechnical Engineering where the Geo-materials are solids.

However, natural soils are complex materials and their behavior is dependent on numerous parameters. Capturing all these parameters in a single mathematical model under certain physical conditions through FEM might become computationally expensive. In order to reduce the complexity, the soil is first divided into multiple elements called substructures, which aide in a more accurate and faster computation(Gulikers, 2018). These are then condensed to a homogenized material model and compiled into a global model that describes the overall behavior of soil. This not only increases the accuracy of the model, but also accommodates various complexities occurring in the soil structure. Since this process requires more parameters with increasing complexity of soil behavior, Machine Learning (ML) is introduced in order to explain the complex behavior using as many simple mathematical models as possible. ML techniques have the ability to generate models using data from past experiences. They help in establishing a relationship between the input variables and the output variables using 'Optimization Algorithms' and thus reducing the errors of approximation.

There are various types of Machine Learning techniques available and every technique has a unique application. In this thesis, the main focus will be on the Artificial Neural Networks (ANN). An ANN is a special type of Machine Learning technique that is similar to a human brain. A human brain consists of neurons which learn from the past data. ANNs too are made up of neurons with an ability to learn from past data(detailed explanation of ANN in chapter 2. ANNs are non-linear statistical models with an ability of approximating non-linear relationships between inputs and outputs and formulate new pattern (Xie et al., 2020). With the help of ANNs, this thesis will focus on deriving a relation between minimal number of Inputs and Outputs for different soil models such as the Linear Elastic Model and the Mohr Coulomb Model and implement the same as constitutive models for Finite Element Analysis.

1.1 Proposed Solution

A soil model can be divided into substructures which represent the element level. Every element level consists of material properties which consists of a material stiffness matrix(homogenised material model), called the material level. This thesis attempts to incorporate a data driven approach in computing the material stiffness matrix. The material stiffness matrix would eventually define the global level stiffness matrix which manifests the soil behaviour.

1.2 Research Questions

The general scientific question of this thesis is: *Could every typical behaviour of the soil be captured using a neural network and further be implemented into the constitutive model?* From the main research question the subsequent questions are as follows:

1. *Could the neural network be trained using anisotropic stress-strain data(simulating natural phenomena) and successfully predict behaviour with standard lab test data? Is the inverse possible too, where natural soil phenomenon could be predicted from training with standard lab test data?*
2. *Could only the strain increments be used as an input to predict the stress increment or are more parameters required as the complexity of behaviour increases?*
3. *What are the standard laboratory tests that must be considered such that the behaviour of every soil model is captured successfully?*
4. *Could the behaviour of simpler soil models be predicted from complex soil models?*
5. *Could soil models like Elasto-plastic model which require computation of linear elastic matrix and plastic flow criterion be modelled using neural networks?*

1.3 Scope of Work

The objective of the thesis is to design a neural network architecture which can predict the behaviour of multiple soil models and explore whether the models can be used to predict the natural phenomenon. The thesis will consist of three parts:

1. Creating datasets of the soil data using two types of lab tests: Standard lab tests such as Triaxial compression test, Triaxial extension test, DSS test and Non-Standard lab test or any other test(s) that maybe required
2. Create a generic neural network and check the model accuracy in predicting natural soil behaviour. If the model is unsuccessful in predicting the soil behaviour, devise an alternative method which could predict the same.
3. Implementation of the neural network in Finite Element Engine such that the constitutive behaviour of soil is predicted, replacing the mathematical computations.

Chapter 2

Background and Literature Review

2.1 Introduction

The focus of the literature study is towards the integration of neural networks in the field of Geotechnical engineering. The first part of the literature study is about geotechnical engineering, applications of geotechnical engineering and various test methods used in geotechnical engineering. This is followed by a the evolution of neural networks in the field of Geotechnical engineering with an initial introduction of neural networks, their basic structure and functioning. Following this, a detailed explanation about the study on the various implementations to integrate the neural networks into Finite Element Analysis, is explained. This is followed by a brief study on the User Material(UMAT) subroutines of various soil models to gain an understanding of the inner workings of the computation of stresses in a soil integrated in an Finite Element Engine. The literature study would also include a brief description of other machine learning models used in prediction of soil behaviour through constitutive modelling. Finally the literature study ends with idea of implementation of the neural networks in Finite Element Engine.

Geotechnical engineering is a branch of engineering which deals with the design, construction, analysis of foundations, slopes, retaining structures, embankments and other systems that are made of or are supported by soil or rock(Tansel and Guoqing, 2021). Due to its heterogeneity, soil is a complex structure that requires a thorough study to understand the behaviour under various physical conditions. Based on the application, the analysis on the behaviour of soil varies. For example:

- **Foundations:**Foundations transmit the load from the structure to the soil. This requires an analysis of various parameters such as the bearing capacity of the soil, settlement, generation of pore pressures etc. Based on the type of foundation, physical conditions of the soil, static and dynamic loads of the structures, the parameters required for analysis vary accordingly.
- **Retaining structures:**Retaining walls serve the purpose of stabilizing the downward movement of soil and rock. The concern for the design of the retaining structure is mainly the lateral earth pressure created by the soil on the structure which could lead to failure of the structure. Hence, a detailed study is required on the density of the soil, the pore pressure in the soil etc.
- **Slope Stability analysis:**In the case of slope stability analysis, the slope stability is determined based on the ratio of shear stress and shear strength. Various factors such as change in pore water pressure, application of external load(seismic activity, construction etc) could lead to a change in the soil strength and eventually cause failure.

For the analysis of soil behaviour for various structures/conditions, computational methods such

as Finite Element Analysis are deemed to be more useful. These methods help in analysing various scenarios with great feasibility by simulate the natural behaviours of soil in various conditions.

2.2 Basics of Finite Element Analysis

The developments made in computation capabilities, mainly Finite Element Method(FEM) have fostered significant developments in numerical methods that are applied in various engineering fields, including geotechnical engineering. Recent developments in Non linear FEM for soil-water coupled problems using sophisticated constitutive models allow us to predict the behaviour of soils and structures with high accuracy under static and dynamic loading conditions(Murakami et al., 2010). FEM is a scheme for solving partial differential equations. In this method, spatially distributed variables such as displacement are discretized with computational grids called "elements" and the discretized variables are given at nodal points of these elements. The procedure for FEM as per Erhunmwun and Ikponmwosa, 2017 is given in 7 steps:

1. *Establishing governing equations and boundary conditions:* In order to generate an approximate solution to a problem, the differential equation for governing the behaviour of the material and the boundary conditions must be defined.
2. *Discretization of domain:* This step includes the subdivision of the entire domain into smaller elements. Typically larger the number and smaller the size of elements, better the approximation of the solution.
3. *Determining the element equations:*The elements once formed, algebraic equations are developed for every individual element. Difference in the algebraic equations from one element to the other would be due to the geometry of the element and the properties of the elements.
4. *Assembling global equations:* Once the element equations are generated, they are put together to form a system of equations for the entire solution domain.
5. *Imposing boundary conditions to the problem:* This step helps in reducing the assembled global equations to a solvable size. It involves substituting some known parameters at some known points into the global equations. In order to do this, specific primary degrees of freedom must be identified.
6. *Solution of global equations:* The system of equations is solved for values of dependent variables at various points on the domain. Depending on the type of problem, there are numerous points at which the solution to the differential equation is approximated.
7. *Presentation of results:* The results are represented in a graphical and/or tabular form.

FEM has been used in several geotechnical applications such as slope stability analysis, coupled hydro-thermal soil behaviour, dynamic soil behaviour etc. Britto et al., 1992 implemented finite element analysis for coupled problem of heat flow and consolidation for disposal of high level radioactive waste buried in the seabed. Savvidou and Britto, 1995 conducted a two-dimensional coupled heat conduction and consolidation analysis of soil barriers subjected to temperature gradients. Yashima et al., 1998 analyzed the behaviour of natural clay at different strain rates and temperatures. A dusty gas model for three gas components was developed by Hibi, 2008 to analyze the movement of gas in soil using FEM, for a soil-vapor extraction and bio-venting system. Several FEM methods have been implemented to simulate soil deformation in certain complex conditions. In Poran and Rodriguez, 1992 a treatment of dry sand under dynamic compaction due to repeated drops of rigid tamper was conducted. The simulation of the impact behaviour of sand was carried out by implementing special computational techniques for re-meshing and reassignment of material properties. This was required to accommodate large deformations and simulate the associated plastic behaviour of this type of sand. The success of a numerical simulation, regardless of the sophistication of FEM, depends on the appropriate constitutive model. It also depends on the input data, i.e., the material data adopted by the constitutive model, the initial boundary conditions, etc. A common method to evaluate the material parameters of any constitutive model is through laboratory experiments on soil samples. The various available laboratory experiments are elaborated in the next section.

2.3 Laboratory tests to study soil behaviour

2.3.1 Constant rate of strain(CRS) test

This test was first described by Hamilton and Crawford in 1959 as a rapid means of determining the pre-consolidation pressure P_c(Selig et al., 1978). In this test, the soil specimen is confined laterally and drainage of water is only allowed from the top. The specimen is then loaded with a constant rate of displacement. The strain rate is chosen such that no significant pore pressures are developed. Theory was developed for the CRS test which led to the expression:

$$C_v = \frac{H^2 \Delta\sigma_v}{2\Delta t u_b} \tag{2.1}$$

and

$$\sigma'_v = \sigma_v - 2u_b/3 \tag{2.2}$$

where H is the length of drainage path in specimen, σ_v is total vertical stress, σ'_v is average effective stress, t is time, u_b is excess pore pressure measured at the base of the specimen and C_v is coefficient of consolidation. The testing equipment looks as in figure(2.1). The test specimen is held inside a stainless steel ring resting on a base. The specimen is then loaded with a constant rate of displacement by moving the piston with a gear driven load frame. During the test the excess pore pressure is measured at the base using a transducer. The vertical load and displacement are measured by an external load cell.

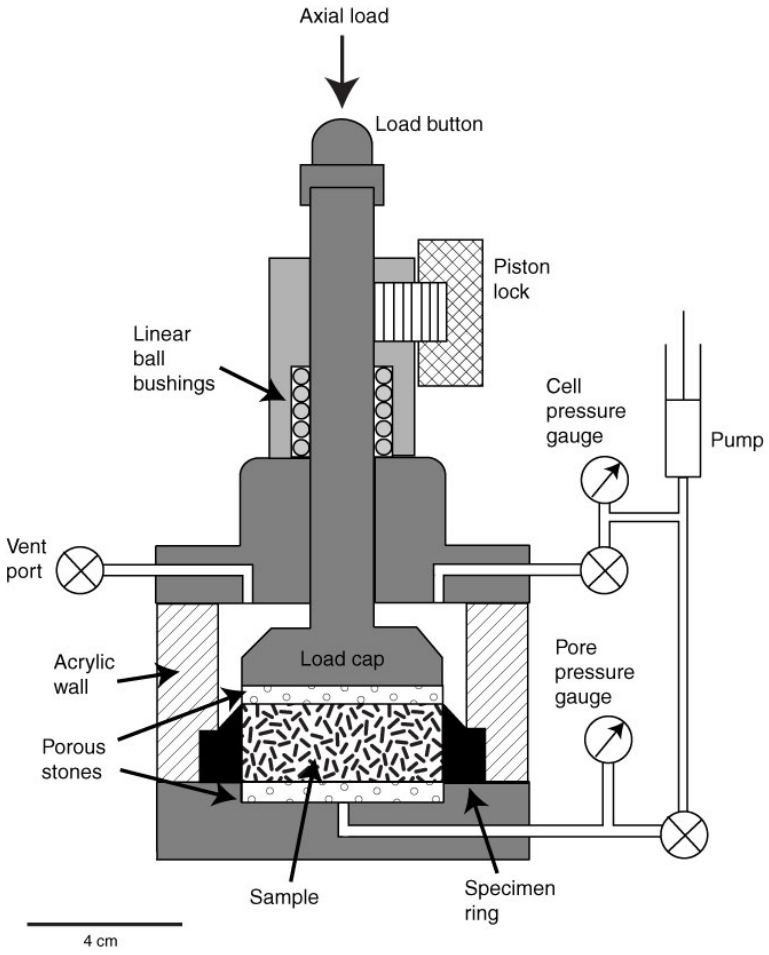


Figure 2.1: CRS test apparatus

Due to the rigid boundary conditions there is no failure that takes place in this test. Also data regarding secondary compression cannot be obtained unless a special procedures are instituted(Gonzalez, 2000).

2.3.2 Triaxial test

Triaxial shear strength on soil measures the mechanical properties of soil such as shear resistance, cohesion, dilatancy etc. In this test, a cylindrical specimen having a length to diameter ratio of 2 is loaded and tested with stress applied in axial direction as shown below:

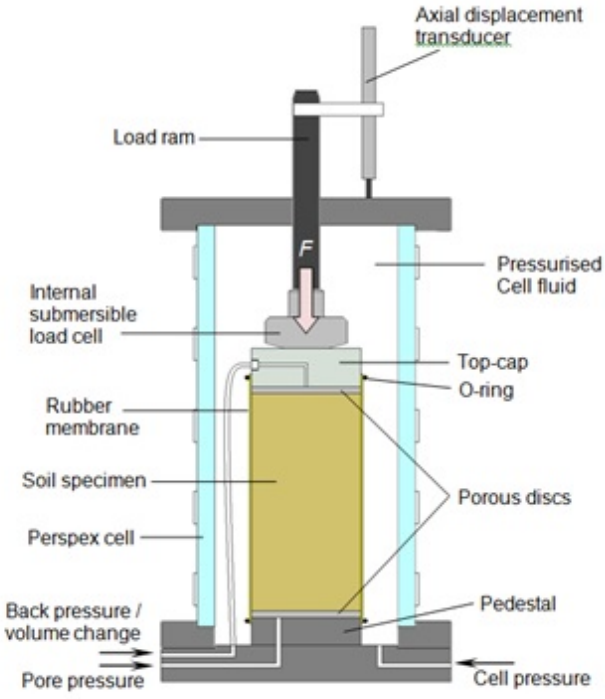


Figure 2.2: Triaxial test apparatus(Rees, 2013)

The sample is subjected to three principal stresses out of which two are applied due to pore pressure within the confining cell. The third stress is the axial stress applied by the apparatus in the vertical direction. The stresse applied to the soil specimen are given in the figure below:

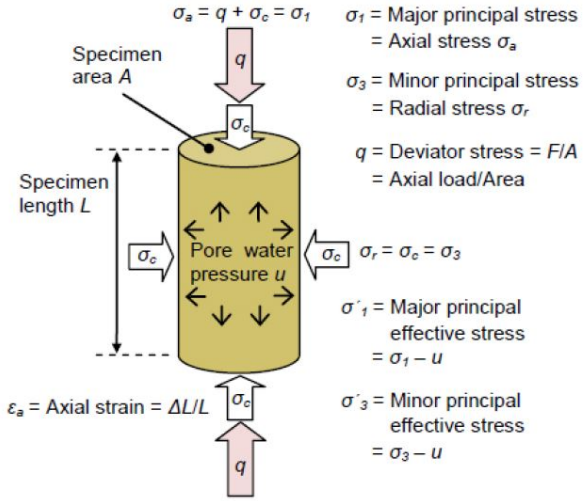


Figure 2.3: Stress state during triaxial test(Rees, 2013)

A confining pressure σ_c is applied by pressurising the cell fluid surrounding the specimen which is equal to the minor principal stress σ_3. The deviator stress q is generated by applying an axial

strain ϵ_a to the soil(Rees, 2013). The deviator stress acts in addition to the confining stress in the axial direction(major principal stress σ_1). The stress state is considered isotropic when $\sigma_1 = \sigma_3$, and anisotropic when $\sigma_1 \neq \sigma_3$(Rees, 2013). The triaxial test can be applied in different variations namely

Unconsolidated undrained test (UU)

In this method, the soil is subjected to pressure under the condition of no drainage. The cell pressure is maintained at a constant value and the applied deviator stress is increased till the sample fails.

Consolidated Undrained Test (CU)

During the application of cell pressure on the sample, drainage is permitted. The deviator stress is applied keeping the cell pressure constant and no provision of further drainage.

Consolidated Drained Test (CD)

The deviator stress is increased by allowing the drainage to occur keeping the cell pressure constant. The loading is applied slowly such that excess pore pressure is not developed within the sample. The prepared specimen is enveloped in the membrane and positioned in the triaxial cell and lateral pressure is applied till the specimen fails. The vertical deformation and the load readings are recorded. The main objective of the test is to determine the values of cohesion and angle of internal friction for which, three different lateral pressure values have to be tested on the sample.

2.3.3 Plane-strain test

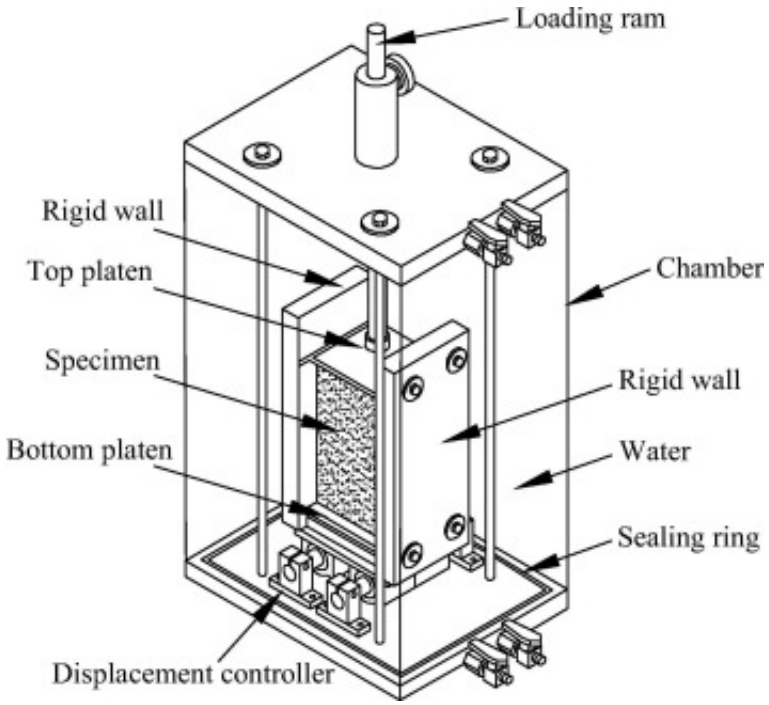


Figure 2.4: Plane-strain or Biaxial test apparatus

In the plane strain state, the strain in intermediate direction (ϵ_2) is constrained thus applying principal stress σ_2. A soil specimen is placed in a cuboidal box which is then applied with load

in axial direction(usually controlled by computer program to adjust loading rates) and the lateral stress is controlled using flexible membranes. The intermediate stress is constrained by injecting water which helps in maintaining the lateral pressure.

2.3.4 Direct shear test

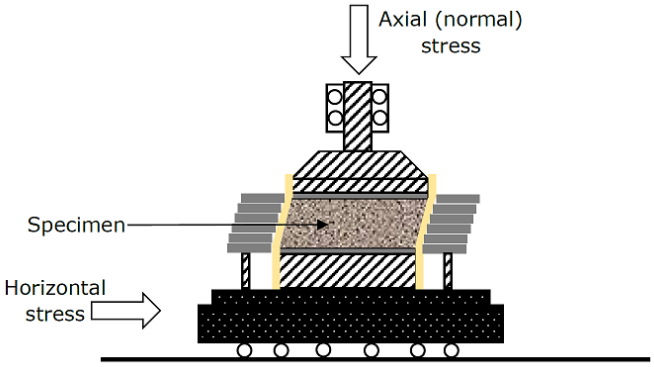


Figure 2.5: Direct Simple Shear test

This test is used to determine the shear strength of the soil. This requires the isolation of soil to only stresses in the shear direction. In this test, initially the specimen is placed inside a shear box which is subjected to an initial confining stress, thus restricting any deformation in principal directions. This is followed by a applying a shear on the sample to cause uniform deformation. In order to ensure uniform deformation, the specimen is constrained by any of the three methods: a stack of metal rings, a wire reinforced membrane or a confining cell pressure(Tech, 2021)

2.3.5 True Triaxial test

In most of the geotechnical field problems, the soil is subjected to principal stresses in three directions. This makes the direct measurement of the stresses in the three directions necessary. A true triaxial test apparatus is designed to measure the deformations undergone by the soil specimen when subjected to principal stresses in all directions. The apparatus for this test has been developing and evolving since long and is still evolving further.

Loading devices with six rigid plates: Developed in 1988, in this apparatus, six sliding rigid plates were used to apply load in orthogonal directions on the cubical soil specimen placed in the system(Yin et al., 2010).

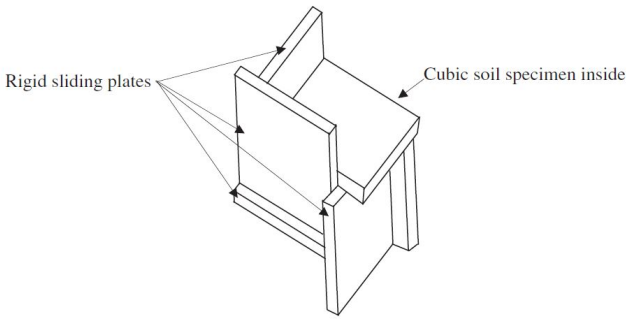


Figure 2.6: True triaxial test apparatus with rigid boundaries(Yin et al., 2010)

The main advantages of this system are:

- Displacement applied by each plate was uniform
- The three principal stresses(σ_{xx}, σ_{yy}, σ_{zz}) could be varied.

The disadvantages of this system are:

- Stresses in the soil specimen may not be uniform
- Drainage tubes were required in the rigid plates, which might increase friction between the soil specimen and the plate.

Loading devices with six flexible plates: Developed in 1967 by Ko and Scott, 1967, in this apparatus, instead of six sliding rigid plates, six flexible loading faces with water-filled rubber bags were used which were used to apply load in orthogonal directions on the cubical soil specimen placed in the system. The box in which the specimen was loaded consisted of 45° bevelled edges and along the edges there was a continuous rectangular groove that was slightly shallower at the corners of the wall. A latex membrane was attached to each wall at the edges. The gap between the wall and the membrane was filled with water. This test was further improved in 1979 by Sture and Desai, 1979. In this test, the soil specimen was floated between six silicone rubber membranes during testing. The specimen was loaded by compressed air applied to these membranes through cylindrical casings.

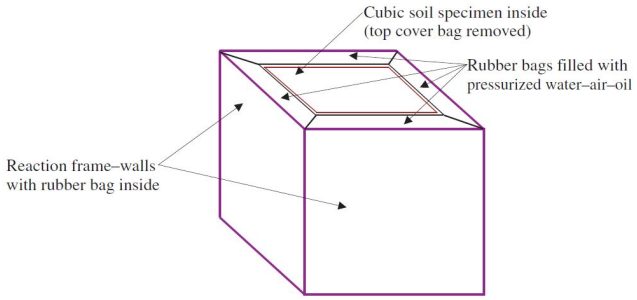


Figure 2.7: True triaxial test apparatus with flexible boundaries(Yin et al., 2010)

The advantages of this system are:

- The stress in the rubber bag could be easily controlled via pressure and the stress applied on the soil surface could be kept uniform on most side surfaces of a cubic soil specimen, except at the bevelled edges.
- The friction between the pressurized bag and the soil specimen was relatively small.

The disadvantages of this system are:

- Strains in the soil specimen may be highly nonuniform, especially at the corner edges(near the bevels).
- Large strains (or displacements) could not be achieved due to the influence of the corner edges.

Loading devices with mixed boundaries: In 1971, Green devised a new true triaxial test apparatus which consisted of both rigid as well as flexible boundaries(Yin et al., 2010). The vertical load on the specimen was applied by a piston thus resulting in major principal stress. The pressure in the cell was provided the minor principal stresses. The intermediate stress was generated by the rigid plates on the left and right side of the soil specimen. A gap was preset between the vertical

and the horizontal plates to avoid interference of plates during loading process. Such gaps could lead to non-uniformity in stress distribution.(Yin et al., 2010)

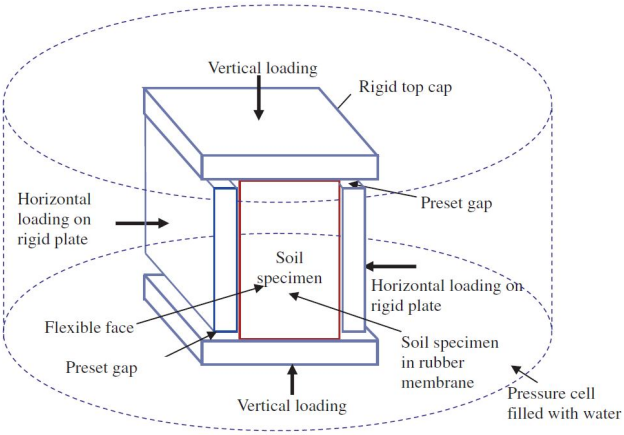


Figure 2.8: True triaxial test apparatus with mixed boundaries(Yin et al., 2010)

The main advantages of this mixed boundary system are:

- The stress can be easily controlled via pressure and it is uniform.
- The friction between the top and bottom plate can be avoided since there is a gap between the corners.

However the disadvantages of this system are:

- Stresses applied on the soil specimen by the top and the bottom plate may be non-uniform, especially at the corner edges(near the bevels).
- There is an interference problem at the corners if the gap is too small or the compression is too large.

The apparatus was further developed to avoid these limitations by Lade and Duncan, 1973. In this system, the soil specimen was enclosed in a rubber membrane and was placed in a large pressure chamber filled with water. The pressure in the cell was used to apply minor principal stresses and the major principal stresses were applied by the top and the bottom plates. The most important design was that the plates were designed such that the vertical plates were compressible(in vertical direction) in nature(made with alternating layers of stainless steel rods and balsa wood). A small gap of 1mm is left between the top plate and vertical plates to avoid friction.

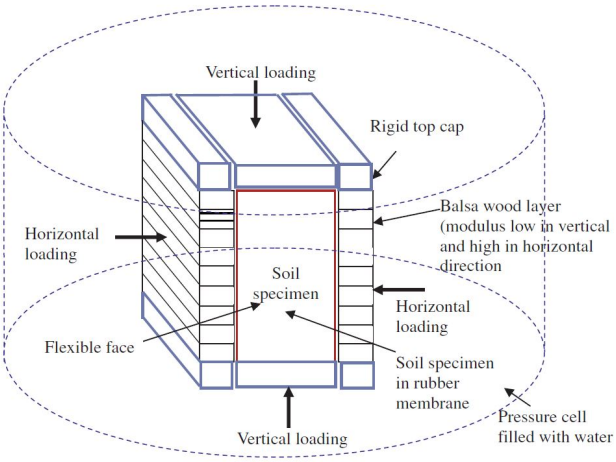


Figure 2.9: Modified True triaxial test apparatus with mixed boundaries(Yin et al., 2010)

The advantages of this system is:

- The stress via pressure in the chamber can easily be controlled.
- The displacement from top and bottom plate are uniform
- There is no issue of corner interference and due to this large strains can be applied as well.

There are certain disadvantages of this system as well

- Although the stainless steel rods were present, there might exist some compression in the material which might lead to some error in the measurement of horizontal displacements.
- The three principal stresses (σ_{xx}, σ_{yy}, σ_{zz}) cannot be fully varied; for example, the direction with the minor principal stress, σ_{zz}, applied using water pressure in the cell is limited to a certain extent and could not be increased to the major principal stress, σ_{yy}(Yin et al., 2010).

However, of all the apparatus developed, this was the most developed system and is used for experiments which can be seen in Abelev and Lade, 2004, Shapiro and Yamamuro, 2003.

2.4 Soil Models based on behaviour of soils

On the basis of behaviour to various physical conditions, soil is classified into various mathematical models or known as constitutive models which aide in understanding typical behaviours such as elasticity, plasticity, hardening etc. Following is an overview on the soil models discussed in this thesis:

2.4.1 Linear Elastic model

A Linear Elastic soil model is the simplest model which based on the Hooke's law of linear stress-strain relation. The generalized Hooke's law can be written as:

$$\sigma_{ij} = C_{ijkl}\epsilon_{kl} \tag{2.3}$$

where σ_{ij} represents the stress tensor, ϵ_{kl} represents strain tensor and C_{ijkl} represent the elastic constants which characterize the elastic behaviour of soil.

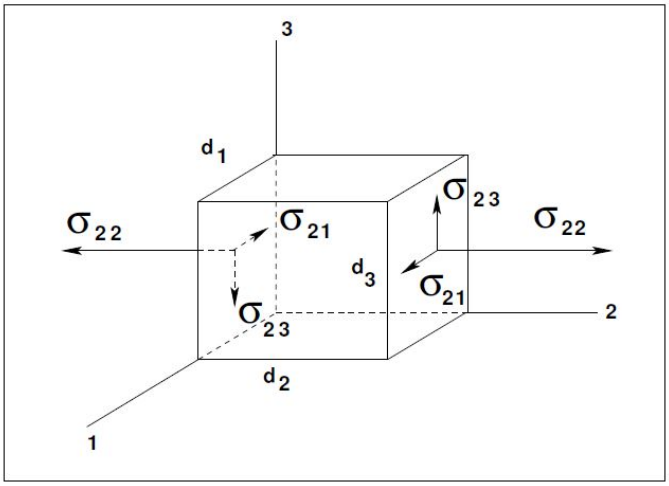


Figure 2.10: Stress components in 3D plane (Kaselow, 2004)

Components of stress tensor with repeating indices for exampleσ_{11} represents normal stress and stress component with different indices is called a shear stress. Consequently,this leads to six stresses i.e., three normal and three shear stresses. If the medium is in static equilibrium, the sum of stresses acting in the three directions equates to zero(Kaselow, 2004). This implies:

$$\sigma_{ij} = \sigma_{ji} \tag{2.4}$$

Similarly, the strains have a symmetric behaviour such that $\epsilon_{ij} = \epsilon_{ji}$. A contracted notation is introduced for stress and strain tensors which is as follows:

$$\sigma_{11} = \sigma_1, \sigma_{22} = \sigma_2, \sigma_{33} = \sigma_3, \sigma_{23} = \sigma_4, \sigma_{31} = \sigma_5, \sigma_{21} = \sigma_6 \tag{2.5}$$

$$\epsilon_{11} = \epsilon_1, \epsilon_{22} = \epsilon_2, \epsilon_{33} = \epsilon_3, \epsilon_{23} = \epsilon_4, \epsilon_{31} = \epsilon_5, \epsilon_{21} = \epsilon_6 \tag{2.6}$$

Therefore, based on the symmetric behaviour of stresses and strains, the elastic tensor C_{ijkl} is reduces to

$$\sigma_p = C_{pq}\epsilon_q, C_{pq} = C_{qp} \tag{2.7}$$

where $p, q = 1, 2, ..., 6$. C_{pq} is called the stiffness matrix. The relation between stresses and strains is represented in the matrix form as:

$$\begin{bmatrix}\sigma_1\\ \sigma_2\\ \sigma_3\\ \sigma_4\\ \sigma_5\\ \sigma_6\end{bmatrix} = \begin{bmatrix}C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16}\\ C_{12} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26}\\ C_{13} & C_{23} & C_{33} & C_{34} & C_{35} & C_{36}\\ C_{14} & C_{24} & C_{34} & C_{44} & C_{45} & C_{46}\\ C_{15} & C_{25} & C_{35} & C_{45} & C_{55} & C_{56}\\ C_{16} & C_{26} & C_{36} & C_{46} & C_{56} & C_{66}\end{bmatrix}\begin{bmatrix}\epsilon_1\\ \epsilon_2\\ \epsilon_3\\ \epsilon_4\\ \epsilon_5\\ \epsilon_6\end{bmatrix} \tag{2.8}$$

The stiffness matrix above represents anisotropic behaviour of soil where there are no planes of symmetry for material propertiesHwu, 2010. If there is one plane of symmetry e.g., $x_3 = 0$, then the stress strain relations reduce to

$$\begin{bmatrix}\sigma_1\\ \sigma_2\\ \sigma_3\\ \sigma_4\\ \sigma_5\\ \sigma_6\end{bmatrix} = \begin{bmatrix}C_{11} & C_{12} & C_{13} & 0 & 0 & C_{16}\\ C_{12} & C_{22} & C_{23} & 0 & 0 & C_{26}\\ C_{13} & C_{23} & C_{33} & 0 & 0 & C_{36}\\ 0 & 0 & 0 & C_{44} & C_{45} & 0\\ 0 & 0 & 0 & C_{45} & C_{55} & 0\\ C_{16} & C_{26} & C_{36} & 0 & 0 & C_{66}\end{bmatrix}\begin{bmatrix}\epsilon_1\\ \epsilon_2\\ \epsilon_3\\ \epsilon_4\\ \epsilon_5\\ \epsilon_6\end{bmatrix} \tag{2.9}$$

Such a material is termed as monoclinic which has 13 independent elastic constants. If a material has two orthogonal planes of material symmetry it can be proved that the symmetry will exist relative to a third mutually orthogonal plane (Ting and Horgan, 1996). Such materials are said to

be orthotropic in nature. The stress strain relations are represented as:

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & 0 & 0 & 0 \\ C_{12} & C_{22} & C_{23} & 0 & 0 & 0 \\ C_{13} & C_{23} & C_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{66} \end{bmatrix} \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \end{bmatrix} \tag{2.10}$$

The stiffness matrix of orthotropic material is reduced to 9 independent elastic constants. The greatest reduction in the number of elastic constants takes place when the material demonstrates symmetric behaviour with respect to any plane(Hwu, 2010). Such a soil is considered to be isotropic in nature. The stress-strain relations can be described using two main parameters: Young's modulus E and Poisson's ratio ν. The stress-strain relation can be represented in the incremental form as:

$$\delta\sigma = D * \delta\epsilon \tag{2.11}$$

where D represents the linear elastic stiffness matrix. The isotropic linear elastic matrix can be represented using the parameters E and ν as follows:

$$D = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & 1-2\nu & 0 & 0 \\ 0 & 0 & 0 & 0 & 1-2\nu & 0 \\ 0 & 0 & 0 & 0 & 0 & 1-2\nu \end{bmatrix} \tag{2.12}$$

Usually linear elastic models are inappropriate to describe highly non-linear behaviour of soils. However, considering piece-wise linear elastic models representing piece-wise stress-strain relations can help in overcoming the non-linearity problem(Berhane, 2006).

2.4.2 Elasto-plastic model

This model represents a linearly elastic perfectly plastic behaviour and is the first order approximation of soil behaviour.

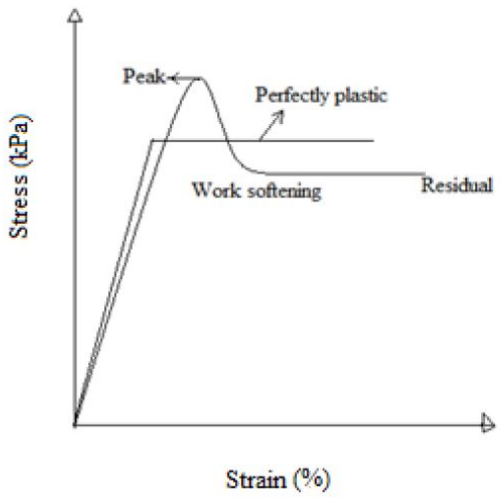


Figure 2.11: Linearly Elastic perfectly plastic behaviour of Mohr-Coulomb model

From the above figure it can be observed that the model has a linear behaviour in the elastic range and has a plastic behaviour in the failure zone. For the linear behaviour parameters like E and ν are sufficient. To explain the failure criterion, additional parameters like cohesion c and friction angle ϕ are required. To model the irreversible volume change during shearing, the

dilatancy angle ψ is also considered. In plastic theory, for the evaluation of the occurrence of plasticity, a yield functions f is introduced which a function of stress and strain. Plastic yielding occurs when $f = 0$. There are several yield criteria developed to describe the non-linear soil behaviour:

Mohr Coulomb model: This model developed in 1773 by Coulomb, derives a relation between shear stress(τ) and normal stress(σ_n) as follows(Yu, 2006):

$$|\tau| = c + \sigma_n tan\phi \tag{2.13}$$

The visual representation of the Mohr-Coulomb model in the deviatoric plane is as follows:

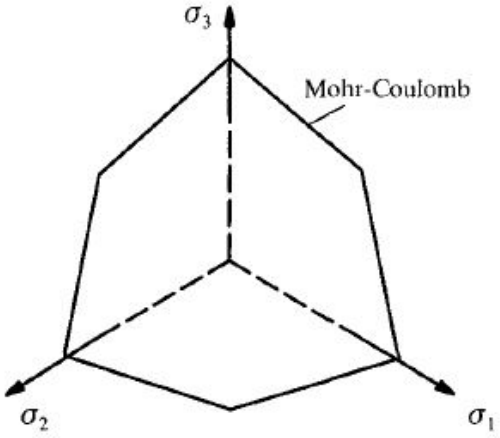


Figure 2.12: Failure criterion of Mohr-Coulomb model(Yu, 2006)

In terms of principal stresses, the yield criterion is represented as:

$$f = \sigma_1 - \sigma_3 - (\sigma_1 + \sigma_3)sin\phi - 2ccos\phi = 0 \tag{2.14}$$

for $\sigma_1 \geq \sigma_2 \geq \sigma_3$ In this model, the failure behaviour is captured under drained conditions. The plastic potential g is the derivative of the yield function with respect to stress. The major disadvantage of this model is that the major principal stress is independent of the minor principal stress which leads to underestimation of yield strength of materials and also disagrees with the results showing influence of minor principal stress on the material strength(Yu, 2006).

Drucker Prager model: This model is a modified version of Mohr-Coulomb model and is widely used in geotechnical applications to predict failure strength.

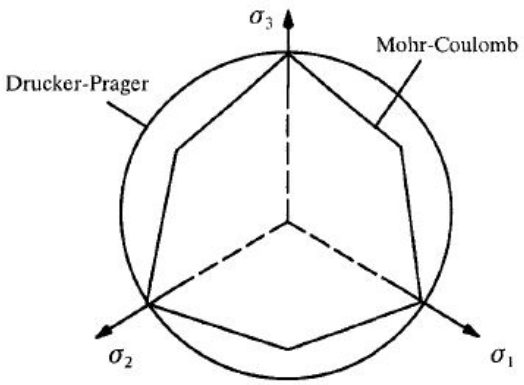


Figure 2.13: Failure criterion of Drucker-Prager model(Yu, 2006)

The failure criterion is represented as:

$$f = \sqrt{J_2} - \alpha I_1 - k = 0 \tag{2.15}$$

where α and k are material constants, I_1 is the first invariant of the Cauchy stress and J_2 is the second invariant of the deviatoric part of the Cauchy stress. Though this model has wide applications in geotechnical analysis, experimental research suggests that having a circular shape on a deviatoric plane does not coincide with the experimental data(Yu, 2006). Hence, care needs to be taken while using this plasticity model.

Tresca model: This model is only applicable in undrained soil conditions(Yu, 2006). The yield criterion for this model is given as:

$$f = \sigma_1 - \sigma_3 - 2S_u \tag{2.16}$$

where S_u is the undrained shear strength. The failure criterion is graphically represented as a regular hexagon in the deviatoric plane:

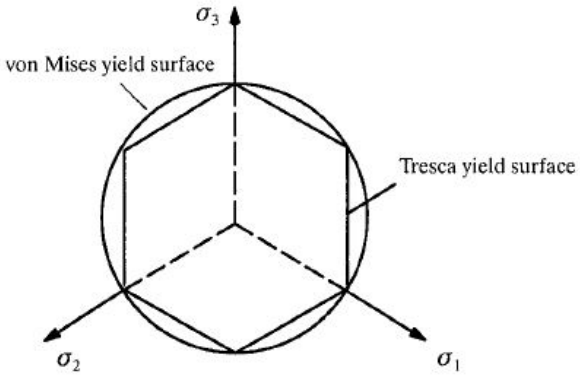


Figure 2.14: Failure criterion of Von Mises and Tresca model(Yu, 2006)

From the figure, it is noted that the plastic potential are not differentiable particularly at certain corner points. These singularities need to be treated separately to avoid singularities(Yu, 2006).

Von Mises model: Developed by von Mises(1913), this model is represented by a circle on the deviatoric plane(in figure 2.14). It is represented by the equation:

$$f = (\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_1 - \sigma_3)^2 - 6k^2 = 0 \tag{2.17}$$

where k is the undrained shear strength of the soil in pure shear. The physical interpretation of von Mises implies that yielding begins when elastic energy of distortion reaches a critical value.(Yu, 2006)

2.4.3 Introduction of Neural networks in Finite Element Analysis

Based on various soil models, the finite element analysis is implemented to study various structures. For soil models representing non linear behaviour such as elasto-plastic models, the computation of stress increments(for change in load) involves mathematical convergence models such as Newton-Raphson method. For non-linear Finite Element Analysis(FEA), this proves to be computationally expensive as well as time consuming. In order to improve the efficiency and accuracy of FEA, an emerging technology called machine learning could be used. Especially the Artificial Neural Network(ANN), which is a versatile machine learning framework, has been proven to be competent in performing various complex tasks such as image recognition to failure analysis of structures. In the following sections, the neural networks are first elaborated, followed by their known implementation in history. This is followed by a literature review and an elaboration on the implementation of Artificial Neural Networks in Finite Element Analysis.

2.5 Neural networks- Structure and Operations

2.5.1 Structure of Neuron

A human brain is an incredibly capable non-linear modelling tool which consists of 100 billion biological neurons. Simple operations are performed by the neurons on receiving an electrical impulse from other neurons, collectively capable of performing tasks such as high-frequency image recognition or calculations. The biological neuron consists of four components: the dendrites, soma, axon and synapses. The neurons are connected to each other in such a way that the synapses of the first neuron is connected to the dendrite of the second neuron and the sequence thus continues. Synapses on being triggered by an electrical signal, passes the signal through the axon and finally the signal reaches the synapses where an electro-chemical contact is made to the next neuron. The brain has a capability to perform predictions, which range from trends in the global financial market to human emotions with a debatable accuracy.

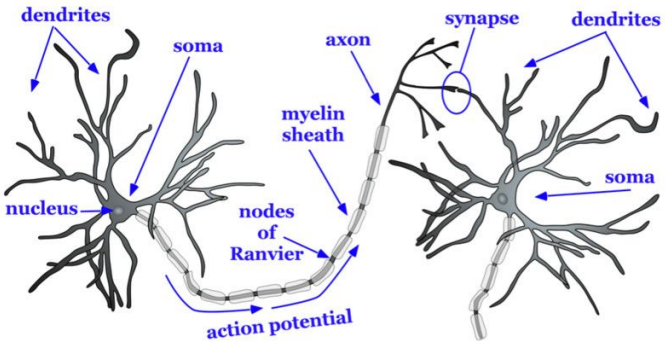


Figure 2.15: Biological Neuron

The operation of human brain and the neurons led to the theory of artificial neurons or nodes which form together Artificial Neural Network(ANN). The operation of the biological neurons was reduced to a mathematical function which was used in the artificial neuron. The artificial neuron receives an input(one or more inputs), that is weighted and summed to further pass through an activation function which is a non-linear function. There are several types of activation functions which include Linear, ReLU, Tanh, Sigmoid etc.

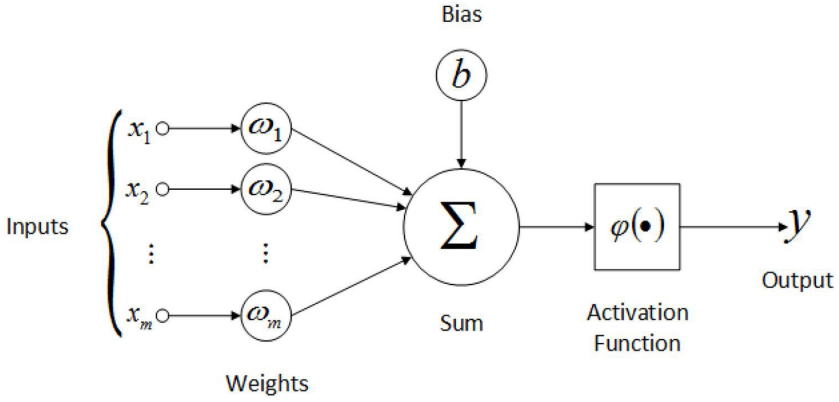


Figure 2.16: Neuron of Artifical Neural Network

In the following sections are a detailed explanations about the various essential components of the Neural network.

2.5.2 Connections in neural networks

For characterisation of complex patterns for a given data-set, there is a requirement for multiple neurons, similar to the human brain. These neurons with the connections form the artificial neural network. Every artificial network consists of a minimum of three layers namely the input layer, the output layer and the hidden layer. For example, for a problem of linear interpolation, the neural network can be simple, with one input layer, one output layer and one hidden layer. Based on the complexity of the problem, the hidden layers can be varied along with the corresponding connections to the input and the output layers.

As mentioned earlier that every neuron unit provides an additive contribution to the unit which it is connected. The total input to a neuron unit k is a weighted sum of separate outputs from each of the connected units added with a bias term θ_k. The equation is represented as follows(Krose & Smagt, 1993):

$$s_k(t) = \sum_j w_{jk}(t) y_j(t) + \theta_k(t) \tag{2.18}$$

In the above equation, w_{jk} refers to the weight of connection and y_j refers to the output from the j^{th} unit.

2.5.3 Activation function

Activation functions represent mathematical functions which define the output of a neuron for a given set of inputs. Each activation function has its advantages and disadvantages and it is the decision of the user to select appropriate functions for the neurons in the network. There are certain properties of the activation function which are to be considered when assigning an activation function:

- Non linearity: Using a non-linear activation function enables the ANN to generalise on non-linear relations in a given data. A linear activation function models a linear trend whereas a non-linear activation function has the ability to predict a parabolic or a tangential trendNielsen, 2019. It can be proven through linear algebra that a network consisting of only linear functions can be reduced to a single neuron which implies that the network cannot predict non-linear behaviour. In contrast, the Universal Approximation Theorem given by Cybenko, 1989 proves that an ANN with a single hidden layer with a non-linear activation function is capable of being a universal approximator for any non-linear relationship(Gulikers, 2018).
- Range: Based on the data, the appropriate activation function has the ability to map the inputs in a neuron to a certain range such as $[-1, 1]$, $[0, 1]$, $[0, \infty)$. An activation function with a limited desired output range helps in improving the stability of the network. The range of the activation function can be determined based on the output range of the mathematical function(Nielsen, 2019).
- Derivative: For training an ANN for using backpropagation, the derivatives of the selected activation functions is required. The learning speed of such methods is directly proportional to the derivative of the activation function(Nielsen, 2019)
- Function close to origin: It is desired but not a requirement that the activation function used approaches the identity function near the origin. According to Sussillo and Abbott, 2014, ANN can be more effective when the activation possess the mentioned property.

Following are some of the most commonly used activation functions:

Linear function

The linear function has a range of $(-\infty, \infty)$ and leads to a constant derivative. The equation of the linear activation function is as follows:

$$f(x) = cx \tag{2.19}$$

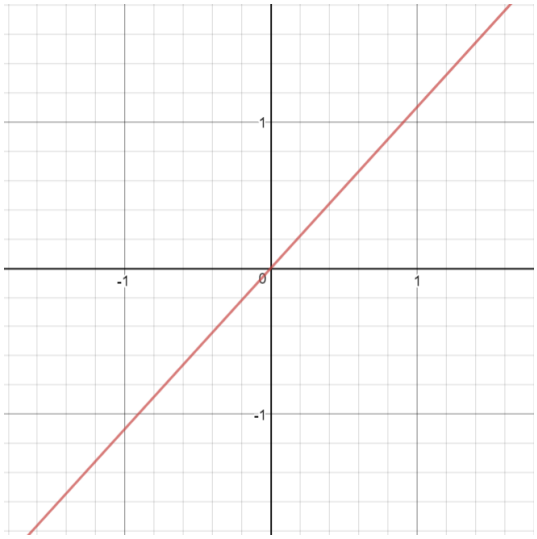


Figure 2.17: Linear activation function

The linear activation function generates outputs which are linearly proportional to the input. Also, the stacking of multiple layers of neurons can be reduced to a single layer while using a linear activation function.

Binary step

This function generates a binary output of either 0 or a 1(an "ON-OFF" switch), which implies the output range of the function is $[0, 1]$. The generated output is 1 when the input was larger than the threshold, otherwise giving an output of 0.

$$f(x) = \begin{cases} 1, & \text{for } x \geq 0 \\ 0, & \text{for } x \leq 0 \end{cases} \tag{2.20}$$

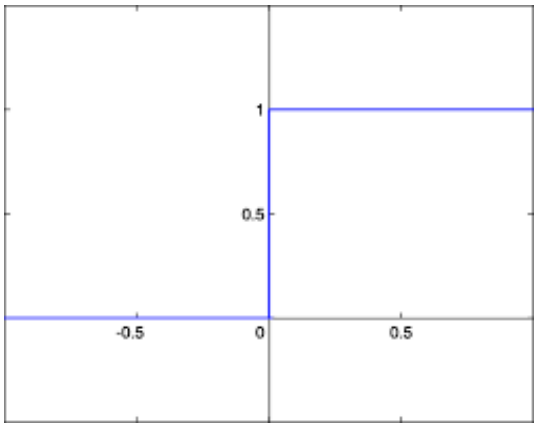


Figure 2.18: Step activation function

The derivative of this function is always 0, except at $x = 0$, where the derivative is undefined.

Sigmoid

Sigmoid function is a very popular activation function and is widely used in machine learning. The output of the sigmoid function ranges between $[0, 1]$. The mathematical formulation is represented as:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.21}$$

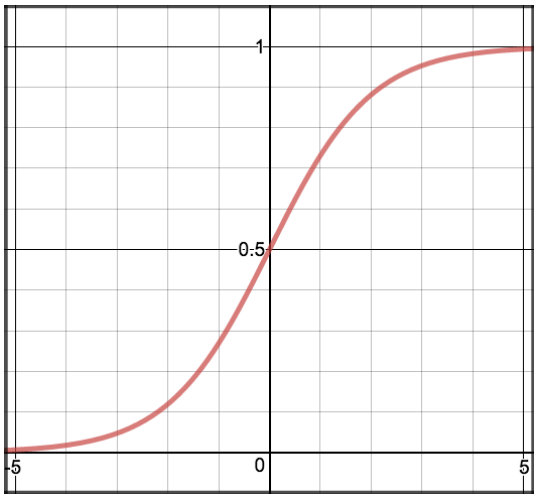


Figure 2.19: Sigmoid activation function

Unlike the binary step function, the derivative of the sigmoid function is non zero and hence can be used for training an artificial neural network(ANN). When the input is negative, the sigmoid function generates outputs close to 0 and when highly positive, it generates an output close to 1. Since, the function produces output limited to values between 0 and 1, this function is used in the output layer of the ANN which helps in normalizing the output data and representing a probability that any of the output classes is true.

Hyperbolic tangent

This function is a scaled version of the sigmoid function and produces an output ranging between $[-1, 1]$.

$$f(x) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}} \tag{2.22}$$

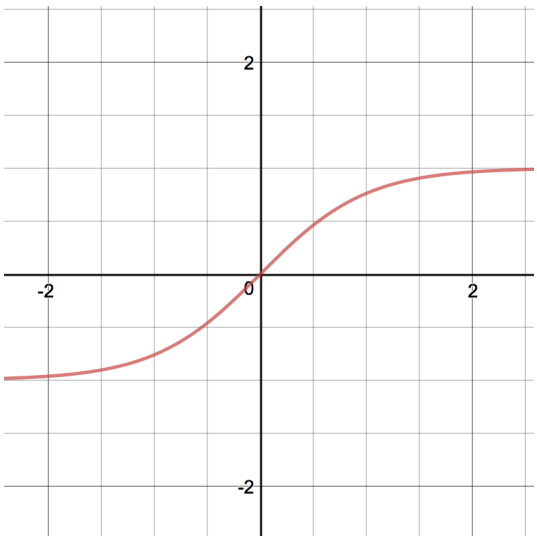


Figure 2.20: Hyperbolic tangent activation function

The mean of the activation function is closer to zero compared to the sigmoid function. Also, the range of the function is twice as large compared to the sigmoid function and features a higher derivative which leads to increase in the speed of learning for the model. If the input becomes larger, the slope approaches to zero, slowing down the learning rate. Similar to sigmoid, tanh function is used as the activation layer in the output layer.

Rectified Linear Unit(ReLU)

ReLU is a popular activation functions used in ANN models. This function is a rectifier with an output range of $[0, \infty)$ and has a well-defined derivative. When the input of the node is lower than zero, the output is zero and if the input is larger than zero, the activation behaves like an identity function.

$$f(x) = max(0, x) \tag{2.23}$$

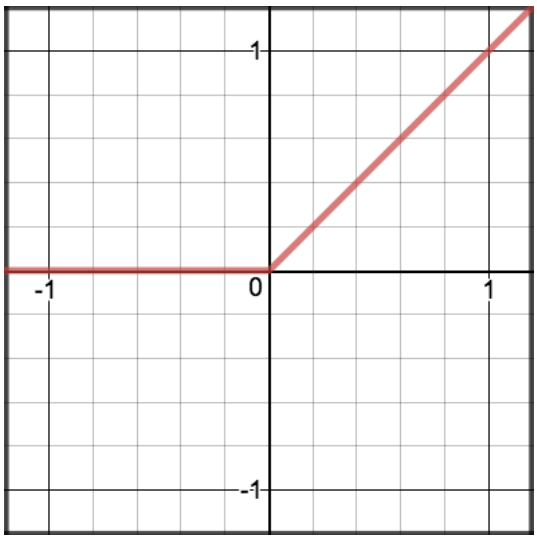


Figure 2.21: ReLU activation function

In the region where the derivative is equal to one, the learning is fast which is one of the main advantages of the ReLU function. When the output is zero, the neuron with the ReLU activation function deactivates and does not participate in the training process as long as it is in that state.

2.5.4 Cost functions and Loss functions

Optimization of ANNs is required to learn trends and patterns from the given data. The amount of optimisation can be measured only by using error functions or cost functions. The purpose of the cost function is to measure the accuracy $\hat{y} - y$ for the obtained prediction $\hat{y}$ in comparison with the actual data y for the corresponding input x. A cost function can be chosen by the user, but there are two assumptions which must be considered while using a cost function(Nielsen, 2019):

- The cost function C should be the average of the cost functions C_x for the individual training examples, x, i.e.,
$$C = \frac{1}{n} \sum_x C_x \tag{2.24}$$
- The cost function should be a function of the ANN output only. This ensures that the input and the hidden layers remain undisturbed during the computation of the cost function.

The cost functions for individual training examples are considered as loss functions or error functions(Khan, 2019). Following are the most commonly used loss functions in the application of ANN. These loss functions consist of notations y_i which is the true output, $\hat{y_i}$ representing the predicted output, for the i^{th} point and n represents the total number of data points in one batch.

Mean Squared Error

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (\hat{y_i} - y_i)^2 \tag{2.25}$$

This quadratic error function is mainly used for logistic regression problems. It is widely used in statistics and can be used to draw a line which features a minimum average squared distance between it and each data point. This cost function depends on the absolute difference of the two values, which implies if y_i and $\hat{y_i}$ are two large numbers relatively close to each other, the computed error is still very high. There is a cost function called the Root Mean Square Error which is a subset of MSE and represents the square root of MSE. It is represented as follows:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (\hat{y_i} - y_i)^2} \tag{2.26}$$

It is widely used in the field of ANN to compute errors where error in the data is highly sensitive.

Mean Squared Logarithmic Error

This function is similar to the previous function but uses a logarithmic expression. The sensitivity in this function is such that two large numbers relatively close leads to a insignificant value of error. If the numbers are small, either of the cost functions can be used to compute the loss. Like MSE, this function is also used in logistic regression.

$$MSL = \frac{1}{n} \sum_{i=1}^{n} (log(\hat{y_i} + 1) - log(y_i + 1))^2 \tag{2.27}$$

Mean Absolute Error

This is a simple linear error indicator that is mainly a fit for linear regression problems. It mainly depends on absolute error between two entities, for example considering two relatively close large quantities, the loss would still be significant. The equation is mathematically represented as:

$$MAE = \frac{1}{n} \sum_{i=1}^{n} (\hat{y_i} - y_i) \tag{2.28}$$

However, this function is still more robust to outliers as larger errors are not amplified.

2.6 Modelling the Neural Network

The Neural network model can be designed using the following steps explained through a flowchart depicted below(Shahin, 2016). A detailed explanation about the sections in the flowchart would be given(excluding choice of optimization method, model inputs as they are beyond the scope of this thesis):

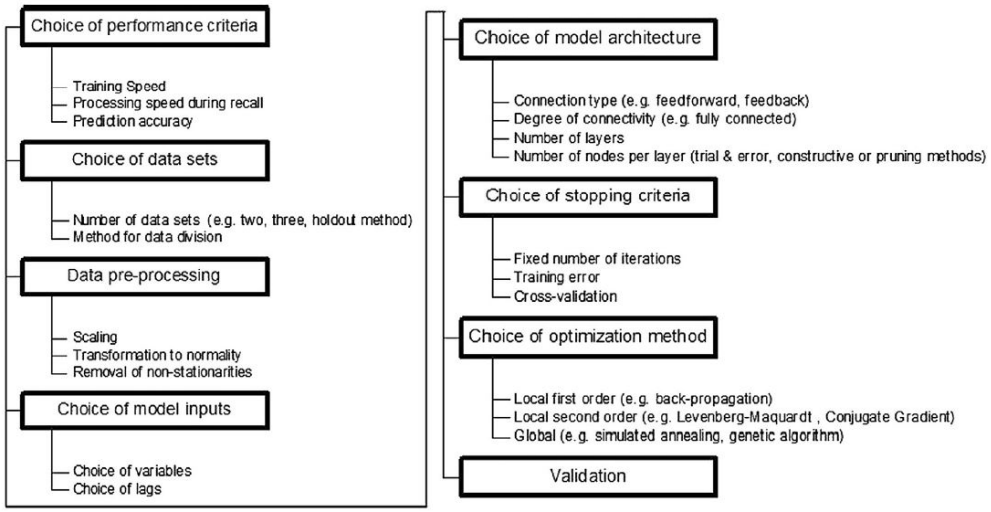


Figure 2.22: Main steps in ANN model development(Shahin, 2016)

2.6.1 Training and Testing Datasets

The datasets created for the model consist of stresses and strains obtain from various laboratory tests. The datasets are subdivided into training and testing datasets before being utilized. The training datasets enable the neural network to learn the trend and pattern of the data by optimizing the weights and biases in the neural network. The testing dataset consists of data which previously was not used to train the network. This testing dataset helps in validating the trained neural network.

Generally, the datasets are split into training and testing data in the ratio $80:20$. However, there are studies which suggest the split of datasets into three, namely training, testing and validation. According to Shahin et al., 2004, the dataset must be divided into the three categories in such a way that training data consists of 55%, testing dataset consists of 25% and validation dataset consists of the remaining 20% of the data. The division of the datasets is arbitrary, but the major aspect to be considered is that the training dataset should be a representative of the entire data on which the network would be tested. This implies that the training dataset must consist of the extreme values of the data. However, if the error from the training dataset is very small, it may lead to the issue of overfitting. Therefore, to check the possibility of overfitting, a separate test dataset is required. There are certain rules which could be followed to avoid the case of overfitting:

- The number of training samples for the network should be greater than or equal to the number of weights present in the network(Rogers & Dowla, 1994).
- Using the method of reguralization where the values of weights in the neural network reduce, leading to the prevention of overfitting.

- Another method which is also considered as a regularization technique is the dropout method. In this method where the weights that are unsuitable in the network leading to undesirable outputs are deleted. This leads to an efficient learning of the model, preventing the issue of overfitting(Nielsen, 2019)

2.6.2 Data Pre-processing

Pre-processing of data includes scaling or normalizing of the data to train the neural network. In many cases, scaling of data is not required as it could lead to loss of information especially in the case of experimental data. But preventing normalisation of data could lead to an instance of inconsistent range of data. In such a scenario, the optimisation process of the model tends to favour larger data present in this dataset, eventually preventing proper convergence(Sola & Sevilla, 1997). Some methods of scaling include:

- Normalization: It involves the process of changing the data to have a unit norm. The objective is to change the data such that they can be described as a normal distribution.

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{2.29}$$

  where x' represents the normalized value. The max value obtained using equation (2.29) is 1 and the minimum is 0.

- Standardizing: The result of standardization or Z-score normalization is that the features are rescaled such that the mean and standard deviation are 0 and 1 respectively.

$$x' = \frac{x - \mu}{\sigma} \tag{2.30}$$

  where μ and σ represent mean and standard deviation respectively.

2.6.3 Hyper-parameters

The hyper-parameters refer to the settings the user has to define before commencing the training process of ANN. Following is the list of hyper-parameters defined for the neural network(Gulikers, 2018):

- Network architecture: Number of hidden layers, neurons per layer, activation functions.
- Data: Training/Testing ratio, normalisation parameters, cross-validation method.
- Optimisation parameters: Learning rate(α), decay rate(λ), number of epochs, batch size, initialisation of weights and biases.

Assuming an optimal combination exists, it is difficult to simultaneously optimize all the parameters. Most of the hyper-parameters are correlated due to which, the optimisation is a tough task to achieve. When tuning a machine learning problem, it could lead to over-fitting or under-fitting.

Over-fitting: When error calculated for a validation dataset increases with a decreasing error for a training dataset, it is considered as fitting to the noise present in the data instead of signal, which in other words is considered over–fitting(Allamy and Khan, 2014). If a network over-fits on a dataset, it is said that the ANN has high variance(Goodfellow et al., 2016). A possible solution to avoid the problem of over-fitting is to increase the number of training data such that the noise in the data does not have an influence on the network. A longer training time also has an influence on the network to lead to over-fitting. Therefore, another solution to reduce over-fitting is to implement an algorithm which stops the training process early(Gulikers, 2018). This is referred to as Early stopping which is elaborated in the next section. One more method to decrease over-fitting is to make use of dropout regularisation.

Under-fitting: In the case of under-fitting, the neural network model is incapable of capturing the non-linear relations in the data. In this case, the ANN is considered to have high bias(Goodfellow et al., 2016). A large error in both training and testing set indicates under-fitting. This usually occurs due to short time for training or low complexity of the network. The only solution to avoid under-fitting is to tune the hyper-parameters. Increasing the training time is the first step to be employed to check the performance of the model. If the problem of under-fitting continues, then the nodes and/or hidden layers should be added. One other possible solution is to optimise the learning parameters and/or change the activation function used in the neural network.

2.6.4 Neural Network Callbacks

A callback is a set of functions to be applied at given stages of the training procedure. It is used to get a view on internal states and statistics of the model during training(Duong, 2019). There are various callbacks available. In this thesis, two main callbacks are discussed, Early-stopping and Dropout.

Early stopping: In early-stopping, the training of the network stops at an optimum level of error for training and validation data. This helps in avoiding the problem of over-fitting and improving the generalization ability of the network. For the early stopping to take place a stopping criterion is required which is usually set to be the validation error or in some cases the patience value. The patience value defines the number of maximum number of epochs the model would train until it reaches a global minimum in validation error.

Dropout: With a limited training data consisting of noise, there are high chances of over-fitting to take place. In order to avoid this, dropout is introduced. In this method, a random number of nodes in the network are deactivated for a certain forward or back-propagation loop. This results in less nodes being trained which eventually reduces the issue of over-fitting.

2.7 Neural network evolution in constitutive soil modelling

In an FE analysis, constitutive model of the soil need to be determined to define the behaviour of each element. The function is to determine a relationship between the stress and the strain in the element, such that it represents the mechanical behaviour of the soil. As stated by Ghaboussi et al., 1991, the material models have been developed in a traditional manner from the time of Robert Hooke as follows:

1. A material is tested and the behaviour is observed
2. The observed behaviour is explained through formulation of a mathematical expression
3. The postulated mathematical model is then used to predict untested stress paths and compare the results with either existing or new experiments
4. The mathematical model is modified accordingly to explain the behaviour of the material

An alternative to this method has been proposed, out of which one of the methods involves the use of Artificial Neural Network where the network builds the material model from experimental or numerical material data. Considering the following FE solution strategy, the ANN should be implemented at the material level such that all the aspects of material behaviour(of interest) are captured, provided the availability of sufficient data.

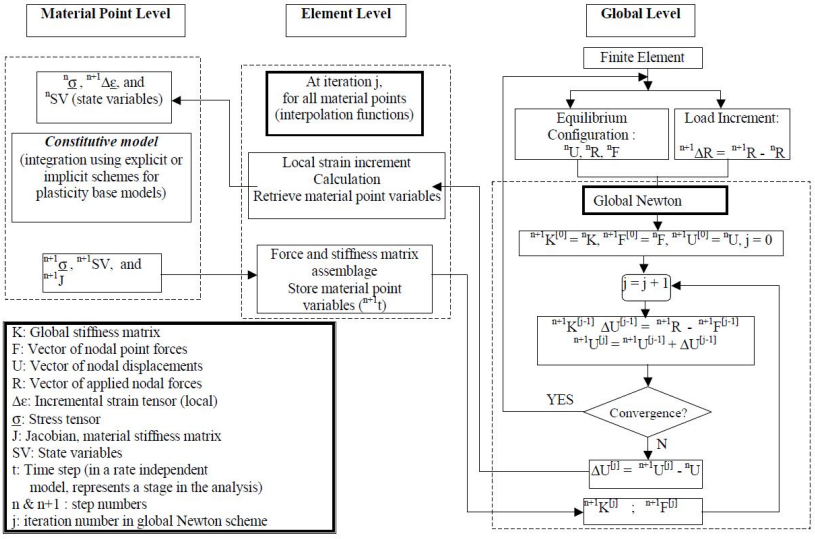


Figure 2.23: Flowchart of implementation of ANN in Finite Element Analysis(Hashash et al., 2004)

Started by Ghaboussi et al., 1991, Artificial Neural Networks were used in modelling constitutive behaviour of soil. This research included the implementation of auto-progressive training of the neural network. The auto-progressive training algorithm implemented, helped in eliminating the challenge of choosing the hyper-parameters for the neural network. This algorithm modifies the network architecture during the training process based on a convergence criteria. The model was quite successful in modelling the constitutive behaviour where the trained data included global load-deflection response measured from a structural test. The neural network was attached to a non-iterative finite element model such that the stress-strain relationship could be extracted to train the Neural network. This idea was later expanded by Ghaboussi and Sidarta, 1998b when the Nested Neural Networks were introduced. This allowed a step-wise method of building and training the model to represent the complex behaviour of soil.

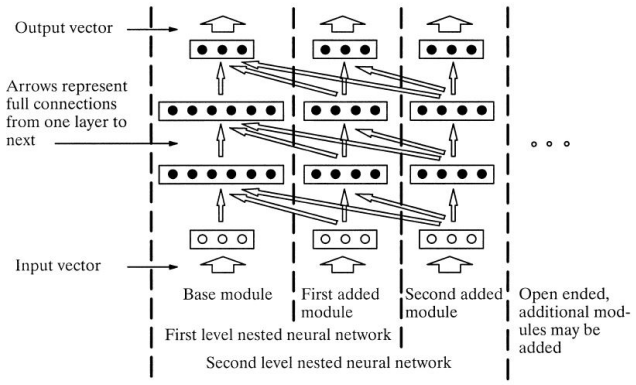
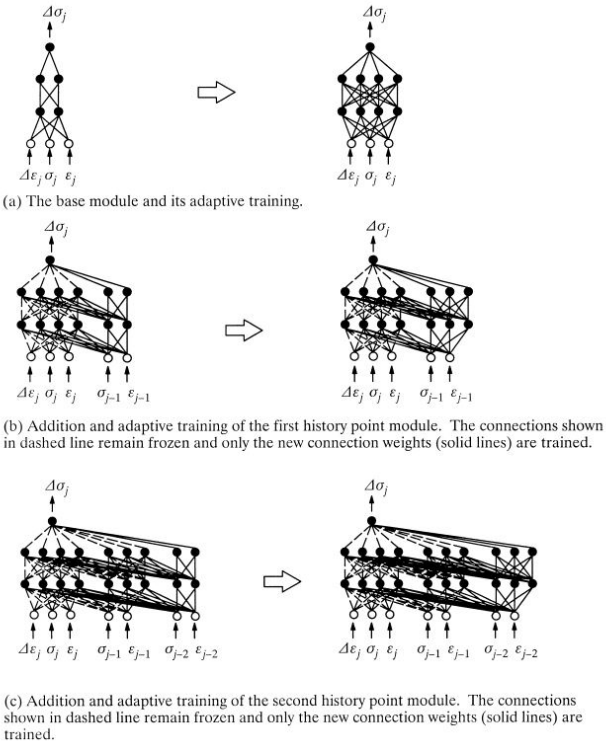


Figure 2.24: Schematic of Nested Neural Network(Ghaboussi & Sidarta, 1998b)

Augmentation of the model was done by adding modules thus forming a higher layer of Nested Neural Network(NNN), where the modules represented a feed-forward neural network. Adding a new module in the network involved freeing the weights of the older module in such a way that the new weights are adjusted.



(a) The base module and its adaptive training.

(b) Addition and adaptive training of the first history point module. The connections shown in dashed line remain frozen and only the new connection weights (solid lines) are trained.

(c) Addition and adaptive training of the second history point module. The connections shown in dashed line remain frozen and only the new connection weights (solid lines) are trained.

Figure 2.25: Evolution and training of Nested Neural Network (Ghaboussi & Sidarta, 1998b)

Training of neural networks was directly performed from the material tests data and then further utilizing the model for analysis of boundary value problems for similar models. This study demonstrated the effect of predicted data which were not included in the training data sets by training a model to predict behaviour of sand under drained and un-drained conditions.

In Ghaboussi and Sidarta, 1998a, the model was expanded by including the history points of stresses to train the model. The model was then used for prediction of strain increments which were not a part of the training data-sets and the performance was analyzed. Zhu et al., 1998 predicted constitutive behaviour of soil using recurrent neural networks. This study involved the prediction of loading and unloading behaviour of soil which was successfully predicted by the model. The model also proved that the use of Recurrent Neural Network was more effective compared to the previous models.

2.8 Implementation of Neural Networks in Finite Element Engine

The above described ANN models take place at a material point level, where the computation of the new stress-strain state is done based on the load increment and the current state. But the drawback lies in the computation of the material stiffness matrix, since the ANN generally omits the stiffness matrix while computing the response. This issue was recognised by Hashash et al., 2004 who derived a relation which aides in explicitly computing the material stiffness matrix through the neural network. The neural network was implemented such that the weights and biases of the neural network could be used to form a material stiffness matrix which then could be integrated into the Finite Element engine. The final derived equation is shown below(derivation of the equation is explained in detail in later sections of the thesis):

$$\frac{\partial^{n+1}\nabla\sigma_i}{\partial^{n+1}\nabla\epsilon_i} = \frac{S_j^\sigma}{S_j^\epsilon}\beta^3 \sum_{k=1}^{NC}\left(\left[\left(1-({}^{n+1}\sigma_i^{NN})^2\right)w_{ik}^{\sigma C}\right]\times\left[\sum_{l=1}^{NB}\left[\left(1-({}^{n+1}C_k)^2\right)w_{kl}^{CB}\right]\left[\left(1-({}^{n+1}B_l)^2\right)w_{lj}^{B\epsilon}\right]\right]\right) \tag{2.31}$$

This neural network derived matrix was compared to a linear elastic model under plane strain loading and was further used for numerical implementation in the analysis of beam bending boundary value problem.

Implementation of neural network in the computation of the stiffness matrix, was done by A. A. Javadi et al., 2003. This involved the computation of the stiffness matrix at every Gauss point i.e, using the neural network to compute the element stiffness matrix. The implementation of the neural network was explained in terms of a flowchart shown below:

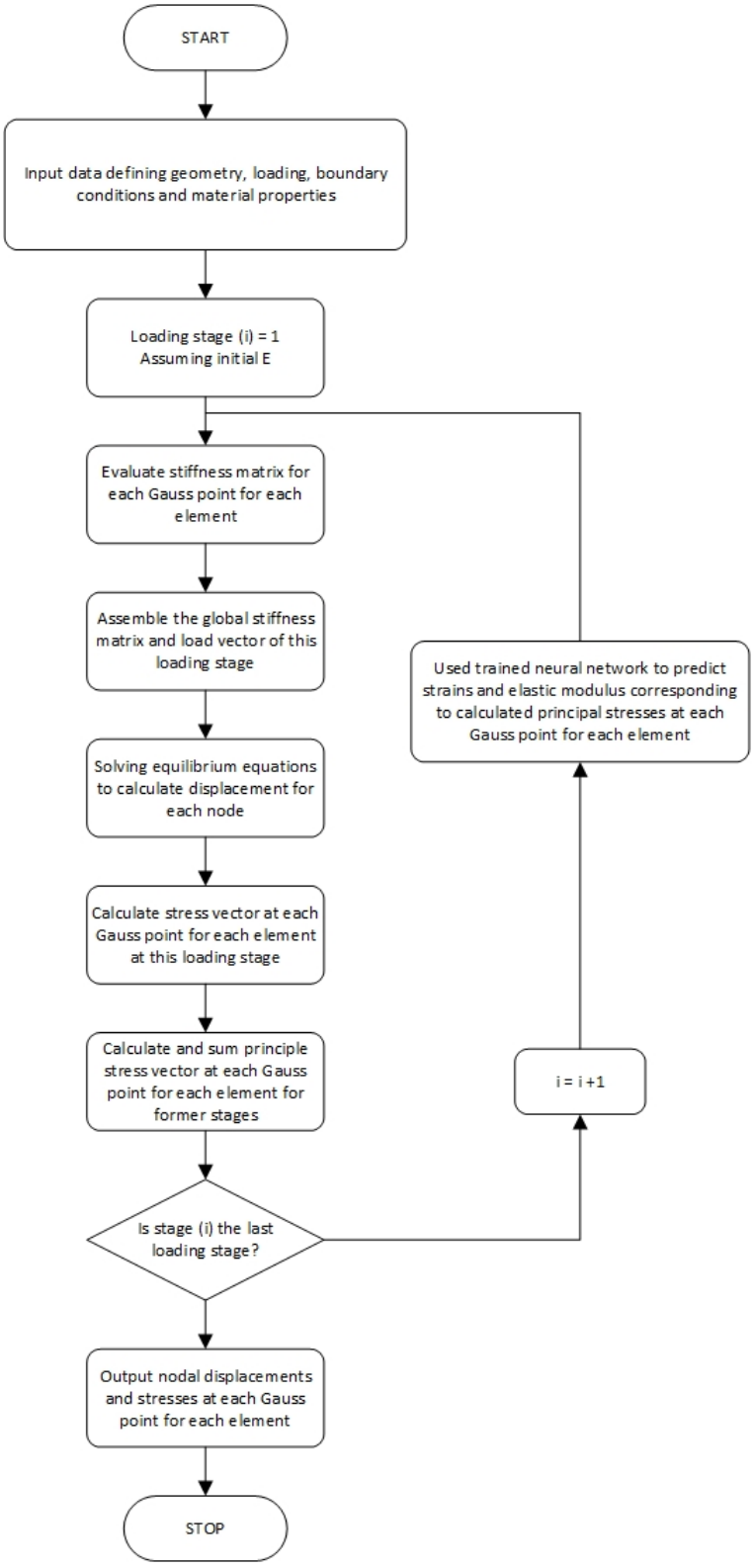


Figure 2.26: FE implementation of Neural Network (A. A. Javadi et al., 2003)

Consequently, the element stiffness matrix aides in the computation of the global stiffness matrix which therefore demonstrate the behaviour of soil/material. The neural network was implemented in the computation of a stiffness matrix in a beam analysis which represented linear elastic behaviour. A second simulation was performed using neural network in an analysis of embankment which consisted of soil following Mohr-Coulomb failure criterion.

According to the Smyrniou, 2018 a neural network was developed to compute the stresses based on strains for three different models, namely Linear elastic , Mohr-Coulomb and Hardening soil models. The main focus of this paper was to evaluate the functioning of various activation functions and behaviour of neural networks with varying activation functions. The neural network was implemented in the element level such that the computation of stiffness is done at every Gauss point. The assumption considered in this paper was the material stiffness matrix to be constant irrespective of the material. Comparing the testing results with the laboratory data, the results were very accurate, for the first two models. In the case of Hardening soil model, the neural network could not be trained to produce accurate results and the inaccuracy occurred at predicting the loading-reloading behaviour of the soil. This implies a requirement for a different approach towards the problem which could also tackle more complex soils such as Cam-Clay, Soft soil creep and Hardening soil models.

Also, Gulikers, 2018 suggests an approach which involves the implementation of the Neural Network into the FE engine through the computation of the material stiffness matrix. According to the thesis, the complexities which occur in the material in the form of structure or the properties could be computed through the implementation of the Neural network in computing the material stiffness matrix which is present in the material level. Integration of the neural network was done into the Finite Element engine by exporting the weights and biases computed from the trained network to a UMAT subroutine where the material stiffness matrix is formulated and integrated into the local stiffness matrix.

2.9 Introduction to subroutines used in FE analysis

2.9.1 User Material subroutines(UMATs)

For a better understanding towards the inner functioning of a finite element engine and the computation of the stresses and strains, a study was done on the UMATs which are used in Finite Element Engines. UMAT abbreviated for User Material Subroutine uses defined failure criterion, which vary according to the soil which subsequently update the material stiffness matrix using a method of convergence called Newton-Raphson method and ultimately compute the updated stresses and strains. The flow chart below gives a brief explanation of the algorithm used in the UMAT subroutine:

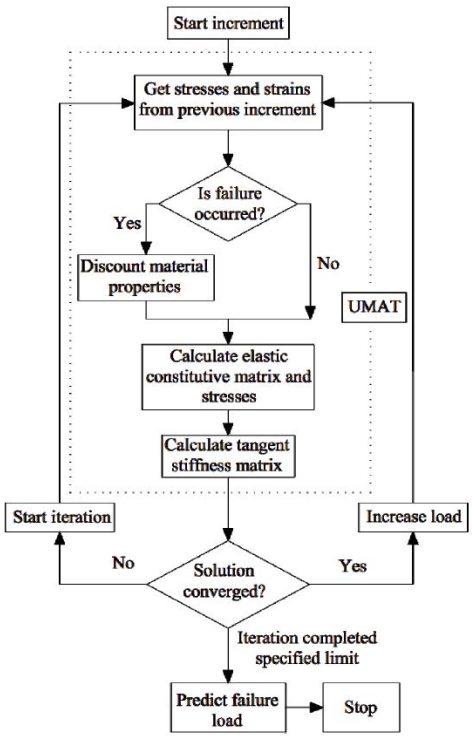


Figure 2.27: Computation of stiffness matrix and stresses in UMAT

Following were the inferences made post reference to UMATs scripted for various soil models.

- In the case of Linear Elastic model, the material stiffness matrix was constant which implies the Hooke's law where stress is directly proportional to strain. The subroutine for this soil model had no change in the material stiffness matrix regardless of the stress imposed.
- In the case of other soil models such as Generalized Mohr-Coulomb, Hardening soil, Norwegian Sand etc., the material stiffness matrix was updated for every change in stress.
- In a soil model such as Hypo-plasticity Sand, there was no explicit involvement of the D-matrix in the subroutine.

The material stiffness matrix computes the element stiffness matrix at Gauss points which integrate to form the global stiffness matrix. The global stiffness matrix gives a broad picture of the behaviour of soil under various physical conditions. Hence, it could deduced from the above information that the material stiffness matrix aides in computing the behaviour of soil under given physical conditions.

2.9.2 User Defined Soil Model(UDSM)

Similar to UMAT which is implemented in ABAQUS, User defined soil model or UDSM is the means to create a constitutive soil model which can be executed in PLAXIS. The UDSM makes use of the material stiffness matrix which integrate to form the global stiffness matrix which are subsequently used for the computation of stresses in the Finite Element model. The UDSM is required to be programmed in FORTRAN which is then compiled into a Dynamic Link Library(DLL) which can then be used in PLAXIS as a constitutive model. For the implementation of UDSM, there are four main tasks which have to be defined to perform the calculations:

- Initialization of the state variables
- Calculation of constitutive stresses which are computed using material stiffness matrix
- Creation of an effective material stiffness matrix
- Creation of elastic stiffness matrix

Implementation of the ANN in UDSM would involve the computation of the stiffness matrix through the neural network and replacement of the effective material stiffness matrix with the former.

2.10 Other machine learning models in constitutive modelling

Apart from Artificial Neural Networks(ANN), one more popular machine learning model used for predicting the soil behaviour is Genetic Algorithm(GA). This involves the Darwinian principal of survival of the fittest. They are stochastic search algorithms which act on a population of possible solutions. The potential solutions are encoded as genes. New solutions are either formed by 'mutating' and 'mating'. Since this algorithm is a selection process, the better solutions are selected to breed and mutate and the worse solutions are discarded.

In Feng and Yang, 2001, this GA was used for behaviour prediction of non-linear material models. The material for which the algorithm was used was a composite material(laminated graphite) which resulted in highly accurate predictions. The subset of Genetic algorithm called Evolutionary Polynomial Regression(EPR) was one model used in the field of Geotechnical engineering for constitutive modelling.

A. Javadi et al., 2009 implemented the method of EPR for constitutive modelling in the Finite Element analysis. The EPR was used to compute the stiffness matrix and was integrated into ABAQUS. The main difference as discussed in the paper, between the ANN and EPR is the output of the neural network. The EPR computes the relation between the input and output in the form of a polynomial equation. This allows in easy integration of the model into Finite Element Engine. This model was used to predict the behaviour of linearly elastic cylinder under plane strain conditions which was accurate compared to standard FEM.

Another application of EPR was presented in Rezania, 2008 where the method of EPR was used to predict the strain softening soil model. The model was tested with lab tests and the model proved to be accurate. Also the model was used in prediction of embankment made of Mohr Coulomb soil subjected to gravity loading. The EPR model was accurate in computing the predictions of the embankment and other numerical examples presented in the paper.

2.11 Summary

The purpose of the literature study was to understand how the Neural networks and the finite element engine function. The chapter began with the various applications of geotechnical engineering. This was followed by an elaboration on Finite Element Analysis and various works related to geotechnical engineering done in this field. Subsequently, various laboratory tests were described, followed by an elaboration on soil models based on the mechanical behaviour of soil in various physical conditions. This was followed by a detailed study about neural networks and implementation of neural networks in FEA. A critical review on the past research and the results of previous research was conducted through the literature study for a promising direction for future research. Most of the literature review failed to deliver a model which could predict non-linear soil models with a successful integration into an Finite Element engine. In conclusion, the objective of the thesis is to develop a more generalized model which could predict non-linear soil behaviour and be implemented in a Finite Element engine.

Chapter 3

Methodology

In this chapter, different methods are used for the creation of ANN to predict soil behavior and implement the same in Finite Element Analysis. Over the course of this chapter, following questions are answered:

1. What are the methods used for creating and implementing the Neural network in Finite Element Analysis and why?
2. How is the data for training and testing the neural network created?
3. What is the evaluation process to verify the accuracy of the created model?
4. What are the system attributes in which the model was constructed and tested?

3.1 Modelling and Implementation of Neural Network

Based on the works of (Hashash et al., 2004), (Gulikers, 2018) and (Smyrniou, 2018), the idea was to create a neural network considering the least number of input variables possible. The neural network would then be used to predict an accurate output and implement the same to create a constitutive model for Finite Element Analysis. In the conducted literature review, it was observed that the input variables considered to predict the soil behavior were Strain(or strain increments) and Stress states to predict stress(or stress increments). Another observation was the consideration of variables only in the principal directions, and in some cases, an additional shear direction. The limitation of this approach is that, the soil is assumed to behave in an isotropic manner in the remaining (unconsidered directions). Hence, to eliminate the assumption and to create a more realistic scenario, variables for the neural network have been considered in six directions (three principal directions and three intermediate(shear) directions). To construct the neural network, a simple to complex approach has been considered. An assumption has been made in this thesis that, there is no pore pressure build-up in the subject soil model.

It has been decided upon to first build a neural network model which could predict a simple Linear Elastic behavior and then to gradually increase the complexity of the network to predict Elasto-Plastic behavior. For the plastic criterion of Elasto-Plastic soil, plasticity models such as Von-Mises and Tresca are applicable only in undrained cases. In the case of Drucker-Prager model which is an available option for drained case, this model over-estimates the stresses during extension (figure 2.13) leading to inaccuracies when compared to experimental data. Therefore, Mohr-Coulomb model has been used as a plastic criterion for Elasto-Plastic soil in this thesis. For the Linear Elastic model, the neural network consisted of strain increment as input and stress increment as output. For the neural network predicting Mohr-Coulomb behavior, two input variables, viz. the strain increment and the stress state were considered and the output of the neural network was stress increment.

Through the course of the thesis, two different methods have been implemented to create the neural network. The objective of the first method was to implement a generic neural network with an ability to mimic the anisotropic behavior of soil. Therefore, for a linear elastic soil model, two types of datasets have been created. In the first type of dataset, the following has been considered:

1. Data is proportionate in every direction (for example $d\epsilon_{xx} = d\epsilon_{zz} = -d\epsilon_{yy}/2$)

2. The dataset consists of two sub-types: one dataset that only considers non-zero values in principal directions, and the other dataset that considers non-zero values only in shear directions. These datasets have been denoted as *Standard datasets*.

For the second type of dataset, the data is expected to demonstrate the anisotropic behavior of soil. Hence, for this dataset, the values for the variables in all directions have been set to be random in such a manner that every direction would impose a different magnitude of load on the soil, thus exhibiting the anisotropic soil behavior. These datasets were called *Non-Standard datasets*. These tests are further explained in detail in chapter 4 followed by the implementation of the methods and their respective results. A visual representation of the classification of datasets is given as follows::

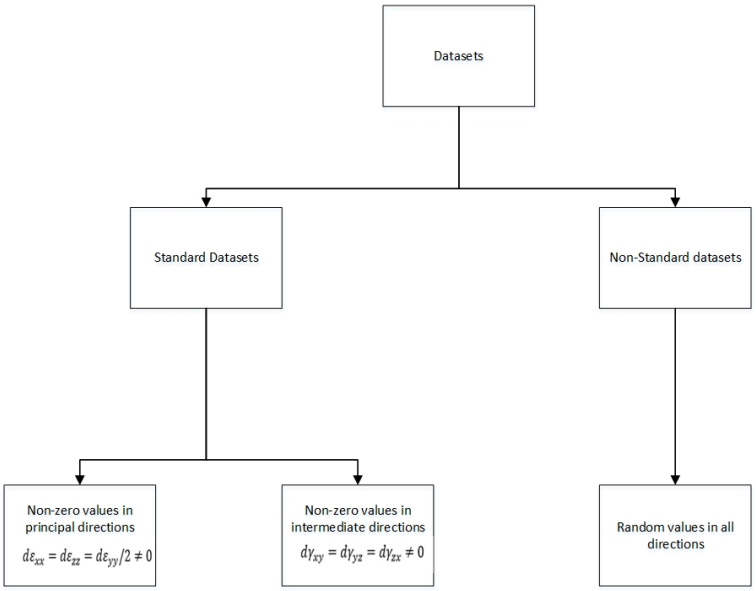
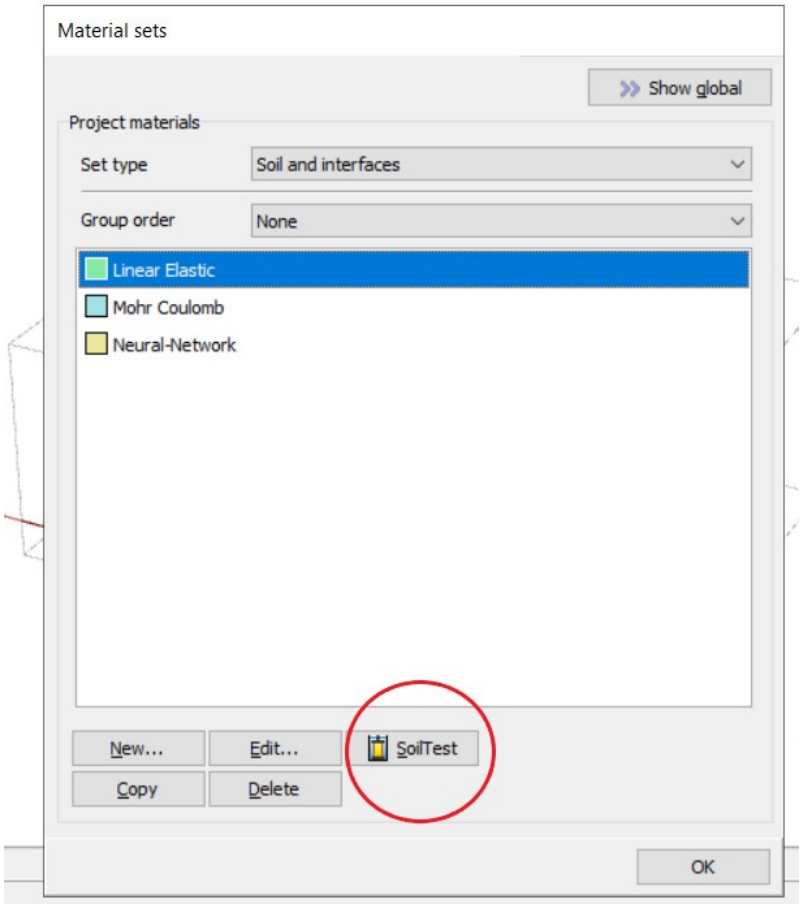


Figure 3.1: Classification of datasets(Standard and Non-Standard) used in Neural network

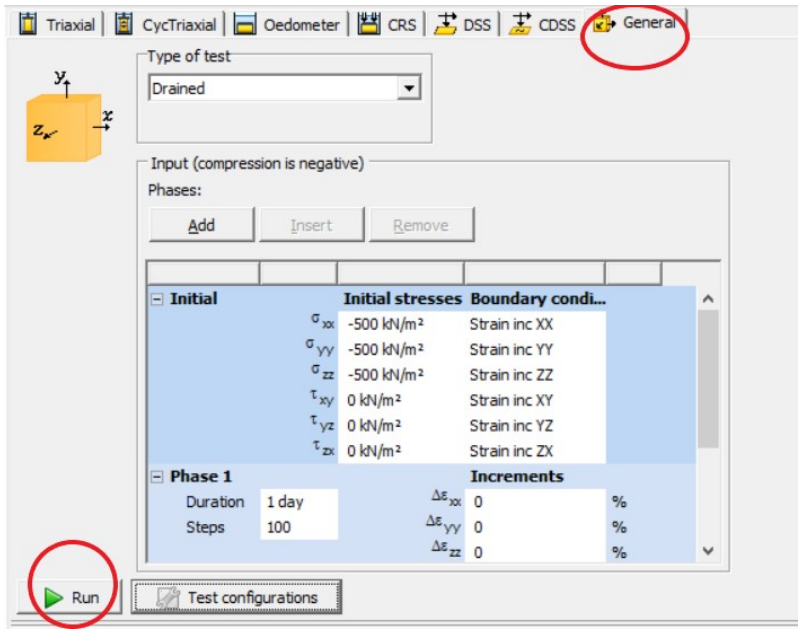
For the second method, the neural network has been constructed with an objective to predict a typical behavior of soil such as linear behavior, plasticity, etc. The motivation for this idea was considered from Smyrniou, 2018 where the author theorized that every soil model is a combination of typical behaviors. These typical behaviors could be expressed using mechanical models such as spring, dashpot and slider. Based upon this idea of mechanical models, neural network has been constructed using certain activation functions (further elaborated in Chapter 5). For the creation of the datasets, standardized lab tests such as CRS test, triaxial test, DSS test etc., have been used. For the training datasets of Mohr Coulomb model, True Triaxial Test has been used, though it is not a standardized test. The reason for using this test is the physical limitations of other tests in establishing a relation between the strain increment in y-direction and stresses in the principal directions. The modelled neural networks have been then tested for accuracy using the metrics that shall be elaborated in later sections.

3.2 Dataset generation and Neural network analysis

The training and testing datasets for the neural network have been generated in PLAXIS using a ".vls" file that was created in Python. In Python program used, the boundary conditions, initial stresses, number of phases and the strain increments were determined to generate . ".vls" files, which have then been stored in the "Soil test" folder created by PLAXIS in the installation directory. In PLAXIS, two soil models- Linear Elastic and Mohr-Coulomb have been created (soil properties in table 4.1 and table 5.4 respectively) in the "Materials" tab. For every material (soil model), soil test has been conducted selecting the "Soil test" tab.



(a) Material tab in Plaxis



(b) Soil test tab

Figure 3.2: Creating a dataset in PLAXIS

In "Soil test", in the "General" tab, the various ".vls" files created have been run to create the stress-strain data. This data has been saved in a ".vlt" format which was converted into a ".csv" file using a Python program. These ".csv" files thus created, have been used as training and testing data for the network.

The datasets have then been compared with each other using histograms which depict different kinds of features in the data. This has been done in order to ensure that the neural network is trained for every extremity possible. The training datasets have accordingly been selected with a caution to not lose the generalization ability of the network. The selected training datasets have been compiled into a single dataframe which has then been used to train the network with a test-train split of 0.20 (the network trains using 80% of the training data and validates the accuracy for the remaining 20%). To then test the accuracy of the network, a single test dataset has been used. The accuracy of the neural network was measured using the Root Mean Squared Error and Mean Absolute Error. The reason for using the Root Mean Squared Error is that, this error method helps when there are large errors present in the network and Mean absolute error aides in understanding the absolute errors in the network and improving the prediction of the neural network. All the neural networks in this thesis have been modeled on the same machine with the following attributes:

Machine	RAM	Storage
Google Colab	12 GB	108 GB

Table 3.1: System attributes of machine used in the thesis

3.3 Finite Element implementation of Neural Network

With the neural network created, using the mathematics referred from Hashash et al., 2004, the Jacobian matrix has been computed from the weights and biases neural network. In order to maintain the continuity of the applied method and the corresponding results, derivation is given in chapter 4 and chapter 6. The Jacobian matrix represents the material stiffness matrix which has then been used to create a UMAT/UDSM for further usage in FEM software like ABAQUS/PLAXIS. UMAT/UDSM is a FORTRAN code which has been generated through Python to create the constitutive model for the respective soils. The Neural Network-implemented constitutive models have then been evaluated for accuracy(by computing the absolute error) using them in a Finite Element Analysis and comparing the results with respective theoretical soil models. The finite element implementation and the results are elaborated in chapter 6.

3.4 Summary

This chapter is a descriptive and a brief discussion about the methods used in this thesis to model a neural network and implementation in Finite Element Analysis. Based on the literature review in chapter 2, a framework has been constructed to build a neural network considering strain increments and stress state as input and stress increment as output. The models have then been considered to be built based on a simple to complex approach. For the elasto-plastic soil model, the Mohr-Coulomb model has been selected as the plastic criterion based on the limitations of other models. Two neural network models have been constructed based on different ideas of approach. The neural network has been modelled to predict the anisotropic behavior of soil to mimic natural phenomena. The second model has been developed based on predicting typical soil behaviors (linear-elasticity, plasticity etc). Based on the models, the datasets have been created accordingly in PLAXIS using python program. The developed neural networks have been evaluated for accuracy using Root Mean Squared error and Mean Absolute Error. The neural networks for respective soil models have then been implemented in UMAT/UDSM to build a constitutive model which has been further used to evaluate the Finite Element Implementation. Though this chapter provides information about the methodology of this thesis, details of these methods will be elaborated in subsequent chapters.

Chapter 4

Initial Approach: Building a generalized Neural network

There have been many mathematical models postulated to explain the soil behavior such as Linear Elastic model, Mohr-Coulomb, Modified Cam Clay, Hardening soil etc. However, in natural conditions, the soil tends to demonstrate a behavior which could be a combination of all these different models, which leads to a complexity in prediction. The objective is to capture the natural phenomenon of soil with the help of Artificial Neural Network by creating a generic network, and further integrating the Neural Network into a Finite Element Engine.

As described in the previous chapter, two types of tests have been considered to construct the neural networks - Standard tests which consisted of proportionate application of strain increments in compression, extension and shear and Non-Standard tests where randomized strain increments(refer chapter 3) have been applied in all directions. The data from these tests have then been used to train the neural networks and also to validate the performance of the networks. To evaluate the generic behavior of the network, the trained networks have been tested with the data the other network was trained with. This is elaborated in detail through the course of this chapter.

This chapter begins with a brief introduction of the methods used to derive datasets from soil tests which have a slight difference compared to the tests mentioned in chapter 2. This is followed by the implementation of the datasets in the neural network model. The corresponding results for the approach are then presented. Following this, is the implementation of the network in a FE model, beginning with the derivation of the mathematics of the jacobian matrix, followed by the results and their interpretation.

4.1 Standard and Non-Standard Soil Tests

4.1.1 Standard Soil tests

There are many tests that are used to understand the behavior of soil. Every test differs from the other based on parameters such as the boundary conditions, hydrological conditions etc. The standard tests used in this thesis are hypothetical tests(derived from existing laboratory tests) to help in understanding the response of soil when one parameter is varied with the remaining parameters either kept constant or increasing at a steady rate. In this thesis, for understanding the behavior of the soil and training the neural network, three tests have been used - compression test, an extension test and direct shear test.

Compression test

In the compression test cylindrical soil or a rock specimen is subjected to confining pressure and loaded axially to fail in compression. Initially, the specimen is loaded in an isotropic condition where the major principal stress (axial load) is equal to the minor principal stresses(cell pressure). As the test initiates, the axial load is increased until the specimen reaches failure. Following is a graphical representation of the stresses in the test.

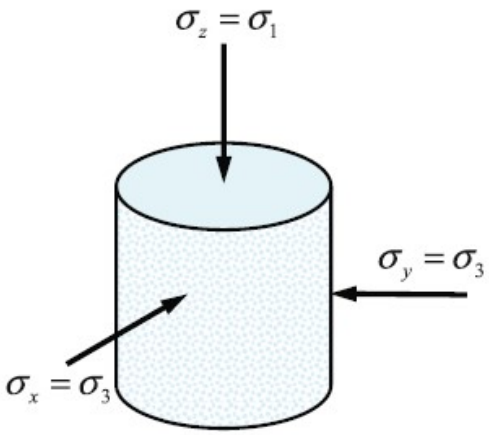


Figure 4.1: Compression test

Extension test

The Extension test is similar to the compression test in terms of the apparatus used but the difference is the subjected loads on the specimen. The stress applied in the major principal direction is horizontal and is higher compared to the stress applied in the minor principal direction which is vertical in this case. The stresses in extension are shown below:

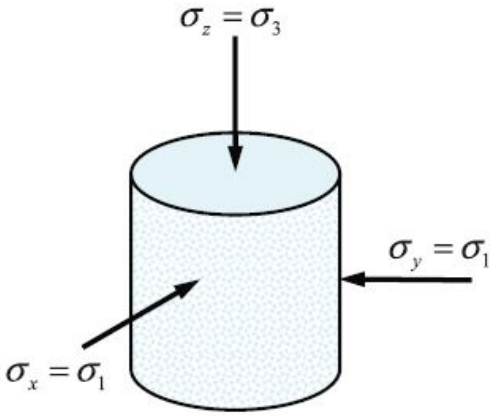


Figure 4.2: Extension test

Shear Test

This test is used to determine the shear strength of the soil. In this test, initially the specimen is placed inside a shear box which is subjected to a confining stress. This is followed by a load applied in the lateral direction which causes the soil specimen to shear.

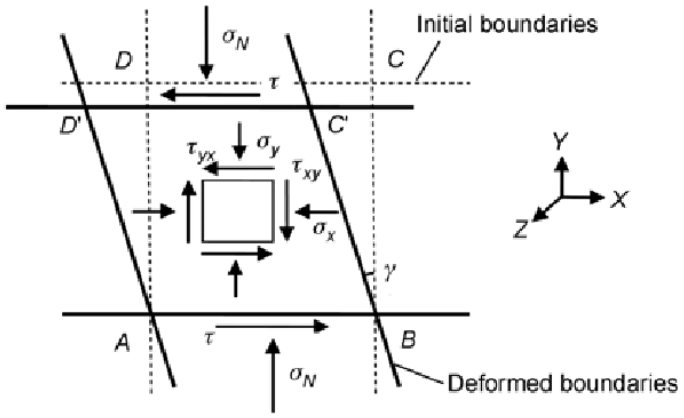


Figure 4.3: Direct Shear test

4.1.2 Non-standard lab test

The above mentioned tests are limited to applying stresses on the soil in a controlled manner which may not be the case in a natural environment. Considering an element of soil in nature, it is acted upon by various loads in every direction, not necessarily with the load of same intensity from every direction.

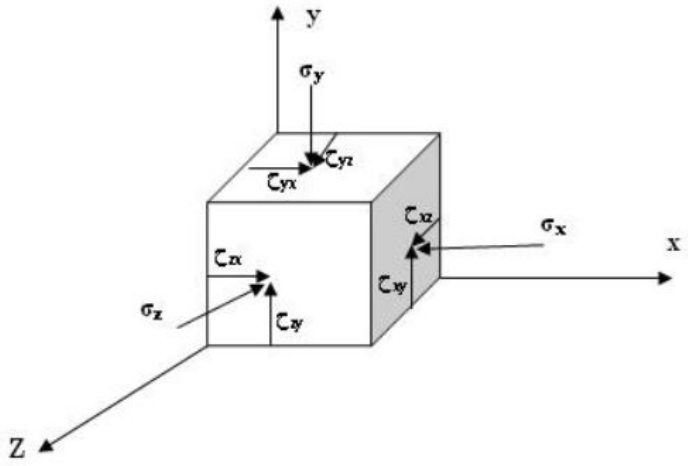


Figure 4.4: Non-Standard test

Since, the objective of the Neural Network is to mimic the nature of the soil, data has been generated where anisotropic loads(randomized strain increments) were applied from all the directions and this generated dataset has been used to train the model. The trained dataset, has then been validated with the datasets of standard lab tests in order to assess the generic performance of the network and to reach a conclusion whether such data can be used for training.

4.2 Data Generation

The data generation for the network has been carried out as elaborated in chapter 3. In order to train the Neural network, it requires the complete behaviour of soil. The strain increments have therefore been applied in multiple phases with a random generation of strain increment in every phase. The soil test consisted of 100 phases of strain-increments for each model. The strain increments were generated randomly(proportionality of strains mentioned in chapter 3) from a normal distribution data with the strain increment ranging from $-5e^{-4}$ to $5e^{-4}$. The initial normal stresses defined for the soil test were $-500kN/m^2$.

With the datasets created, the next step involved the modeling of neural network architecture to predict

the soil behavior. As mentioned earlier, the approach has been to initially model a simple network and build the model with more complexities with increasing complex behaviors of soil models. Initiating with a simple soil model, Linear Elastic Soil Model has been considered.

4.2.1 Neural network for Linear Elastic soil

Linear Elastic soil is considered to follow Hooke's law which states the direct proportionality between stress and strain. Hence there is a linear relationship between the stresses and the corresponding strains. Following were the soil properties used to generate the datasets.

$E(MPa)$	ν
0.5	0.3

Table 4.1: Soil Properties for Linear Elastic model

Based on the above data, the initial idea was to consider strain increments as input to obtain stress increments as output. This was followed by using a python script to generate datasets for standard as well as non-standard tests. For the standard tests, compression, extension and shear tests were considered. The data for the isotropic shear tests generated in PLAXIS consisted of values in the $xy - direction$, the same value which was then replicated to yz and zx directions eventually to train the model. This dataset was then appended with the compression and extension tests' datasets to form a dataframe consisting of the response in all the six directions. The dataframe was then normalized together to maintain the consistency and prevent instablity in the training of the model. The normalization of the dataframe was done for the range between -1 and 1 using *MaxAbsScaler*(Pedregosa et al., 2011). Apart from the standard tests, the non-standard test is also considered where five datasets were generated with the initial stresses implemented in all the six directions and the strain increment also applied in all the directions. The response of stress were recorded for the datasets and a separate network with the same architecture is trained for the non-standard tests. For the training dataset, 3 laboratory tests of each type(i.e., extension, compression and DSS tests) were concatenated to form a single training dataset consisting of $3 * 3 * 10000$ data points. This was followed by creating a validation dataset and a testing dataset by concatenating the datapoints derived from one soil test each which would create a validation dataset of $1*3*10000$ data points each. The neural networks have been subsequently compared with the models of prediction to evaluate the difference in their responses. In the neural network, the input and output have a size of 6 elements (strain increments and stress increments respectively). It also consists of one hidden layer (used for the model) comprising of 6 neurons with a linear activation function. Further, material stiffness matrices have been computed for both the networks from the weights and biases of the networks and compared with the theoretical stiffness matrix (2.12). The formulation of the material stiffness matrix is explained in the next section.

4.2.2 Stiffness matrix calculation

In Finite Element analysis, the consistent material jacobian matrix is also known as the incremental material stiffness matrix. It represents the deformation of the material at a material integration point, thus defining the constitutive behaviour of the soil at each iteration. The mathematical expression for the jacobian is as follows:

$$C = \frac{\partial \delta \bar{\sigma}}{\partial \delta \bar{\epsilon}} \tag{4.1}$$

where $\delta\bar{\sigma}$ represents the Cauchy stress tensor, and $\delta\bar{\epsilon}$ represents the incremental strain. For FE constitutive models, $\delta\epsilon$ is the input and the matrix $\mathbf{C}$ represents the known material law. The multiplication of the two is the output which is the incremental stress $\delta\sigma$. Also, the material stiffness matrix is used to compute the global stiffness matrix by first assembling to form the elemental stiffness matrix.

Following the paper of Hashash et al., 2004, an attempt was made to derive the equation for the stiffness matrix for the neural network model. For the neural network model to predict linear elastic behaviour, weights of the network would sufficient to form the material stiffness matrix. Hence, the biases were deactivated. Therefore, for a given node of a hidden layer H_i, and given input node X_i, weight W_{ij} and σ representing the activation function, the output from the hidden layer could be expressed as:

$$H_i = \sigma(W_{ij} X_i) \tag{4.2}$$

Considering the linear elastic model, with the input strain vector ϵ_i. The strain vector is scaled as follows:

$$\epsilon_i^{NN} = \frac{\epsilon_i}{S_i^\epsilon} \tag{4.3}$$

where S_i^ϵ represents the normalisation parameter for the strain vector For the hidden layer, B_i, the equation is as follows:

$$B_i = Linear\left[\beta\left[\sum_{j=1}^{N\epsilon} W_{ij}^{B\epsilon}\epsilon_j^{NN}\right]\right] \tag{4.4}$$

where $W_{ij}^{B\epsilon}$ = connection weight between the input node ϵ_j and the hidden layer node B_i
β = constant for the Linear activation function

For the output stress vector, σ_i^{NN},

$$\sigma_i^{NN} = Linear\left[\beta\left[\sum_{j=1}^{NB} W_{ij}^{B\sigma}B_j\right]\right] \tag{4.5}$$

De-normalising, the stress vector implies:

$$\sigma_i = S_i^\sigma \sigma_i^{NN} \tag{4.6}$$

where S_i^σ represents the normalisation parameter for the stress vector. The jacobian is represented as:

$$\begin{aligned} {}^{n+1}J &= \frac{\partial\left({}^{n+1}\Delta\sigma\right)}{\partial\left({}^{n+1}\Delta\epsilon\right)} \\ &\Longrightarrow \\ \frac{\partial^{n+1}\Delta\sigma_i}{\partial^{n+1}\Delta\epsilon_j} &= \frac{\partial\left({}^{n+1}\sigma_i - {}^{n}\sigma_i\right)}{\partial^{n+1}\Delta\epsilon_j} = \frac{\partial^{n+1}\sigma_i}{\partial^{n+1}\Delta\epsilon_j} \end{aligned} \tag{4.7}$$

From equation (4.1),

$$\begin{aligned} \frac{\partial^{n+1}\sigma_i}{\partial^{n+1}\Delta\epsilon_j} &= \frac{\partial^{n+1}\sigma_i}{\partial^{n+1}\sigma_i^{NN}}\frac{\partial^{n+1}\sigma_i^{NN}}{\partial^{n+1}\Delta\epsilon_j} \\ &= \frac{\partial^{n+1}\sigma_i}{\partial^{n+1}\sigma_i^{NN}}\frac{\partial^{n+1}\sigma_i^{NN}}{\partial^{n+1}\epsilon_j^{NN}}\frac{\partial^{n+1}\epsilon_j^{NN}}{\partial^{n+1}\Delta\epsilon_j} \end{aligned} \tag{4.8}$$

However,

$$\begin{aligned} S_i^\sigma &= \frac{\partial^{n+1}\sigma_i}{\partial^{n+1}\sigma_i^{NN}} \\ \frac{1}{S_j^\epsilon} &= \frac{\partial^{n+1}\epsilon_j^{NN}}{\partial^{n+1}\Delta\epsilon_j} \end{aligned} \tag{4.9}$$

Hence, from equation (4.8),

$$\frac{\partial^{n+1}\sigma_i}{\partial^{n+1}\Delta\epsilon_j} = \frac{S_i^\sigma}{S_j^\epsilon}\frac{\partial^{n+1}\sigma_i^{NN}}{\partial^{n+1}\epsilon_j^{NN}} \tag{4.10}$$

From equation (4.10),

$$\frac{\partial^{n+1}\sigma_i^{NN}}{\partial^{n+1}\epsilon_j^{NN}} = \sum_{k=1}^{NB}\left(\frac{\partial^{n+1}\sigma_i^{NN}}{\partial^{n+1}B_k}\frac{\partial^{n+1}B_k}{\partial^{n+1}\epsilon_j^{NN}}\right) \tag{4.11}$$

For a linear activation function, the derivative implies:

$$\frac{\partial Linear(f(x))}{\partial x} = \beta\frac{\partial f(x)}{\partial x} \tag{4.12}$$

Hence,

$$\frac{\partial^{n+1}B_k}{\partial^{n+1}\epsilon_j^{NN}} = \beta w_{kj}^{B\epsilon} \tag{4.13}$$

Similarly,

$$\begin{aligned} \frac{\partial^{n+1}\sigma_i^{NN}}{\partial^{n+1}B_k} &= \frac{\partial}{\partial^{n+1}B_k}\left(\beta\left[\sum_{j=1}^{NB} w_{ij}^{\sigma B}\left({}^{n+1}B_j\right)\right]\right) \\ &= \beta w_{ik}^{\sigma B} \end{aligned} \tag{4.14}$$

Hence, the Jacobian for the neural network can be expressed as:

$$\frac{\partial^{n+1}\Delta\sigma_i}{\partial^{n+1}\Delta\epsilon_j} = \frac{S_i^\sigma}{S_j^\epsilon}\beta^2 \sum_{k=1}^{NB} w_{ik}^{\sigma B} w_{kj}^{B\epsilon} \tag{4.15}$$

Considering, the value of $\beta = 1$, this implies

$$\frac{\partial^{n+1}\Delta\sigma_i}{\partial^{n+1}\Delta\epsilon_j} = \frac{S_i^\sigma}{S_j^\epsilon} \sum_{k=1}^{NB} w_{ik}^{\sigma B} w_{kj}^{B\epsilon} \tag{4.16}$$

4.3 Results

Considering the defined neural network with the strain increment as input, the stress increment as output and one hidden layer with a linear activation, the neural network was trained separately for two types of datasets i.e., the model is trained and tested for datasets produced from standard lab tests and also from datasets produced from non-standard lab tests.

4.3.1 Prediction of network for Standard test

The neural network was trained with the datasets fromed from the Standard tests. Following was the plot for predictions obtained for strain vs the stress in the principal directions. The model was trained with a batch size of 16 and is trained for 3 epochs. It can be observed that the values of prediction coincided(superimposition of graphs) with the actual values.

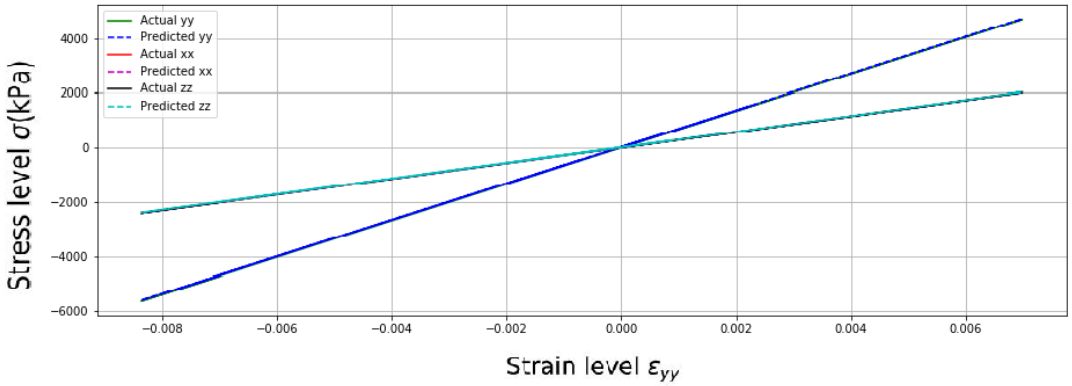


Figure 4.5: Predictions for Standard lab data

To critically analyse the accuracy of prediction, the RMSE(Root Mean Square Error) computed for the predictions is found to be 0.00055635. The error value being very small, the network can be considered to have a good accuracy.

4.3.2 Non-Standard lab test datasets

A separate neural network with the same specifications as above was trained for non-standard lab test where anisotropic loading was applied from all the directions and the results were analyzed. It can be observed that the model was able to learn the trend from the data and make an accurate prediction(in figures below).

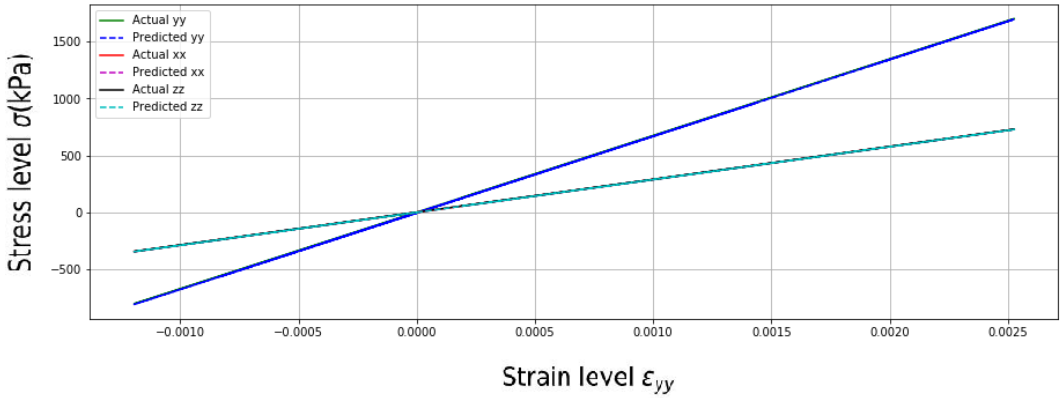


Figure 4.6: Predictions for Non-Standard lab data

The RMSE computed for the predictions to analyse the accuracy of the model is found to be 0.0012467. The error for this model was almost twice of the network trained with standard tests. The increase in error was due to the inconsistent variation of strain increments in the data in non-standard test. However, the error for prediction being a small value, the model can be considered to have a decent accuracy.

Furthermore, to check the generic behaviour of the model, the network trained with non-standard data, was tested with a standard test dataset. Following were the results obtained.

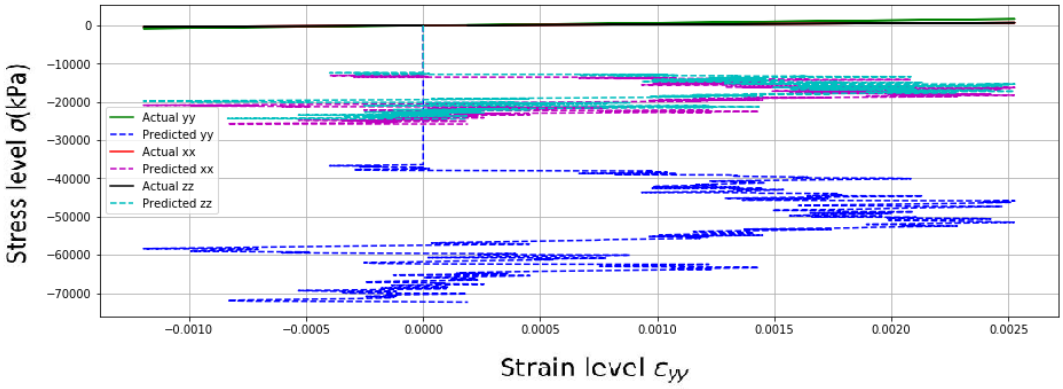


Figure 4.7: Predictions for Standard lab data post training with Non-Standard lab data

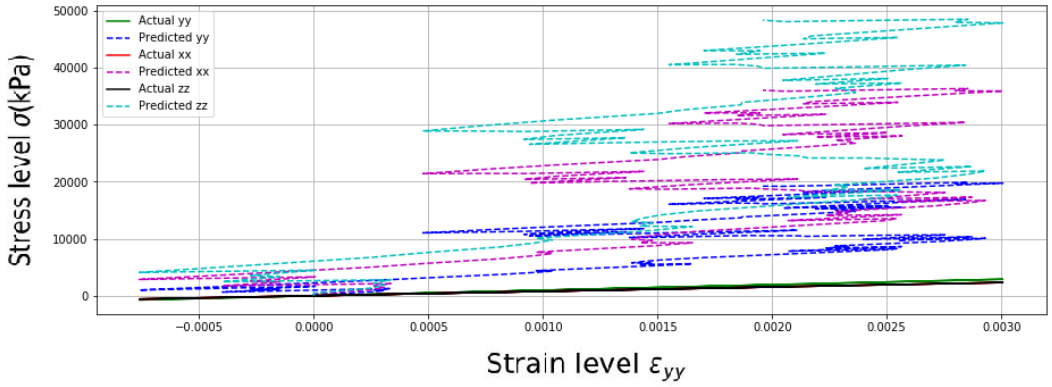


Figure 4.8: Predictions for Non-Standard lab data post training with Standard lab data

From the results it is evident that the data predicted for the standard lab datasets is not accurate when trained with non-standard lab data. The reason for the inaccuracy is that the training data with which the model was trained is different from the data that the model was expecting. Also, the opposite has been checked where the network trained with standard lab tests has been checked with non-standard lab data and the results are similar - the prediction was erroneous. Following is the computed RMSE for the both the results:

Data trained with	Data for Prediction	RMSE
Non-Standard lab	Standard lab	1.54739
Standard lab	Non-Standard lab	2.069328

Table 4.2: RMSE for accuracy of two models

4.4 Finite Element Implementation

Next step involves the implementation of the neural network in the Finite Element engine. Therefore, for this purpose, a UMAT(FORTRAN) was created through a python code which would feed the weights of the neural network into the corresponding layers in the script. The UMAT has been created for the neural network trained with standard tests due to the least errors obtained in the prediction. Using, equation (4.16), the material stiffness matrix has been computed. The execution of the ANN UMAT has been done by implementing the UMAT in program 56($p56$) extracted from Smith et al., 2015b, and the outputs from both UMATs are given below:

```
The integration point (nip= 1) stresses are:
   Element     x-coord     y-coord     z-coord
   sig_x       sig_y       sig_z       tau_xy      tau_yz      tau_zx
      1      0.2500E+00  0.5000E+00 -0.5000E+00
 0.1113E-03 -0.1515E-02 -0.9086E-02  0.2734E-04  0.8176E-03  0.3856E-04
      2      0.2500E+00  0.5000E+00 -0.1500E+01
 0.3205E-03 -0.5395E-03 -0.6521E-02 -0.1129E-04  0.8439E-03  0.4784E-04
      3      0.2500E+00  0.1500E+01 -0.5000E+00
-0.1046E-03 -0.1185E-02 -0.8993E-03  0.8984E-05  0.8106E-03 -0.2672E-04
      4      0.2500E+00  0.1500E+01 -0.1500E+01
 0.1726E-03 -0.8936E-03 -0.2680E-02 -0.2097E-04  0.1263E-02  0.2637E-04
      5      0.2500E+00  0.2500E+01 -0.5000E+00
 0.2987E-04  0.6990E-04 -0.8058E-04 -0.1787E-04 -0.5362E-04  0.1042E-04
      6      0.2500E+00  0.2500E+01 -0.1500E+01
 0.6259E-04 -0.4663E-03 -0.2570E-03 -0.3323E-05  0.5502E-03 -0.4988E-06
```

Figure 4.9: Results for the default UMAT

```
The integration point (nip= 1) stresses are:
   Element     x-coord     y-coord     z-coord
   sig_x       sig_y       sig_z       tau_xy      tau_yz      tau_zx
      1      0.2500E+00  0.5000E+00 -0.5000E+00
-0.4206E+04  0.1043E+05 -0.9362E+02 -0.2314E+04  0.9283E+04  0.3199E+04
      2      0.2500E+00  0.5000E+00 -0.1500E+01
 0.3853E+03 -0.1076E+03 -0.2404E+03 -0.6410E+02 -0.6773E+03  0.7381E+02
      3      0.2500E+00  0.1500E+01 -0.5000E+00
-0.2742E+03  0.4543E+03  0.4373E+03 -0.2391E+03  0.3207E+03  0.4159E+02
      4      0.2500E+00  0.1500E+01 -0.1500E+01
-0.5130E+01  0.4408E+02  0.7974E+01 -0.2051E+02  0.5614E+01  0.1425E+02
      5      0.2500E+00  0.2500E+01 -0.5000E+00
-0.2317E+01  0.4633E+01  0.1024E+02 -0.5214E+01 -0.1795E+01 -0.5380E+00
      6      0.2500E+00  0.2500E+01 -0.1500E+01
-0.9958E+00  0.2061E+01  0.1205E+01 -0.8811E+00  0.1516E+01  0.3738E+00
```

Figure 4.10: Results for the ANN UMAT

It can be observed that the data predicted by the Neural Network is different from the data computed by the default UMAT of the program. For a general overview towards the performance of the ANN, the mean absolute error between the outputs has been computed:

Stress	Mean Absolute error
σ_{xx}	812.3238
σ_{yy}	1840.446
σ_{zz}	131.7879
τ_{xy}	440.6342
τ_{yz}	1714.987
τ_{zx}	554.927

Table 4.3: Absolute error between stress computation using ANN UMAT and $p56$ UMAT

The values for the default UMAT did not coincide with the values obtained from those computed from the Neural Network. This implies that the stiffness matrix for the linear elastic neural network model was not able to produce values similar to the theoretical isotropic linear elastic matrix. Mathematically, there are infinite number of solutions which can be used to derive a relation between strain and stress increment and the obtained solution could help in accurate predictions of the data. However, it is not necessary that the obtained solution of the matrix is the same as the theoretical linear elastic matrix. Therefore, this led to the inaccuracies when the neural network was implemented in FEM.

4.5 Limitations of the approach

The approach of creating a generic neural network led to inaccurate prediction of the data. Hence a critical evaluation was conducted to find the limitations of the approach which could be rectified to obtain a successful neural network model with the ability to predict soil behaviour. The limitations are enumerated as follows:

1. Considering the practical application of the neural network model, the tests used for training the neural networks namely, standard and non-standard tests cannot be replicated in a real-life scenario. Also, in non-standard tests, the randomized stresses applied in all the directions does not necessarily have to demonstrate anisotropic behaviour. Similarly, isotropic behaviour is not a guaranteed implication of proportionate stresses applied in standard tests. Therefore, standardized tests such as triaxial test, DSS test, CRS test will have to be considered for training the network.
2. To analyse the generalization ability of the network, using two different kinds of data would not result in accurate predictions due to different stress paths formed. Hence, a different approach such as introducing noise in the data will have to be employed to analyse the generalization ability of the network.
3. Though the network has been able to produce accurate predictions, in the case of finite element implementation of the neural network, the weights of the neural network formed the linear elastic stiffness matrix which is different from the theoretical matrix. Due to the mathematical possibility of infinite solutions, the neural network will have to be constructed using constraints for weight updation of the neural network model.

Based on the limitations of the neural network, a new approach needs be employed to create a successful neural network model which can also be implemented in FEM.

4.6 Summary

This chapter started with an elaborate description about the standard and non-standard tests used in training the network. This was followed by a description of the neural network architecture considered to predict linear elastic soil behaviour. The trained models were then tested for prediction and it was observed that both the models were sufficiently accurate in prediction. However, the network trained with standard tests had a better accuracy compared to the network trained with non-standard tests. The reason was due to the inconsistent variation in the data from phase to phase, thus decreasing the efficiency of learning the behaviour accurately. The trained networks were then tested for accuracy by testing the standard data trained network with non standard data and vice-versa. It was observed that the prediction of the models was erroneous due to different stress paths created through the data. This answers the first research question *Could the neural network be trained using anisotropic stress-strain data(simulating natural phenomenon) and successfully predict behaviour with standard lab test data? Is the inverse possible too where natural soil phenomenon could be predicted from training with standard lab test data?*

The more accurate model(model trained with standard data) was then implemented in FEM to evaluate the accuracy and the results obtained consisted of significant errors. The errors occurred due to the inefficiency of the model to form a material stiffness matrix which would coincide with the theoretical material stiffness matrix. This was followed by a critical analysis of the network to understand the reasons for inaccurate predictions and drawback of the approach. From this approach, it was observed that implementing anisotropic behaviour of soil is not possible using lab tests. Therefore, the idea of simulating anisotropic behaviour of soil is not practical at this stage further research would be required to model this behaviour. In conclusion, the approach used in this chapter was not successful and based on the drawbacks of the approach, a better neural network model needs to be created to predict the soil behaviour accurately.

Chapter 5

Alternate Approach: Building a Neural network based on typical soil behaviour

The approach to implement a "generic" neural network in the earlier chapter was inadequate, since the network was unable to predict the anisotropic behavior of soil. Based on the limitations of the previous approach to the solution, changes have made in the implementation of the neural network model to learn the soil behavior. As mentioned previously in chapter 3, the idea for the neural network has been based on making use of specific components to explain typical behaviors of soil. The components were then to be integrated to derive the soil model accordingly. In this chapter, soil models of linear-elastic and Mohr-Coulomb shall be defined using the components along with their implementation and their assessment with respect to the accuracy of the models. Another change compared to the previous approach are the laboratory tests used to generate training and testing datasets. In this approach, already existing laboratory tests have been used such as CRS, DSS, Triaxial test etc. The models shall be then tested for their generalization ability, followed by sensitivity analysis to understand the weightage of variables in the neural network.

5.1 Components for various behaviours

A complex behavior of soil can be expressed from typical behaviors like elastic behavior, plastic behavior, creep, etc. These constitutive behaviors can be explained using mechanical components such as the spring, slider and dash-pot(Smyrniou, 2018). A complex soil model can be developed by a combination of these mechanical components. The present approach is to use the Neural network as material models in order to capture the typical behaviors, which shall subsequently help in predicting of complex soil behaviors. In the following sections, typical behaviors(elasticity and plasticity) shall be studied and a Neural network shall be modeled for each behavior.

5.1.1 Linear Elastic model

As per the description in chapter 2, linear elastic soil follows Hooke's law. Hence this behavior can be explained using a reversible spring which would define Elasticity.

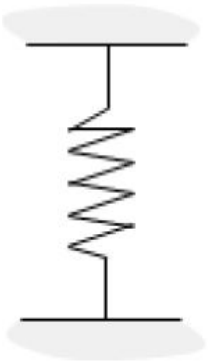
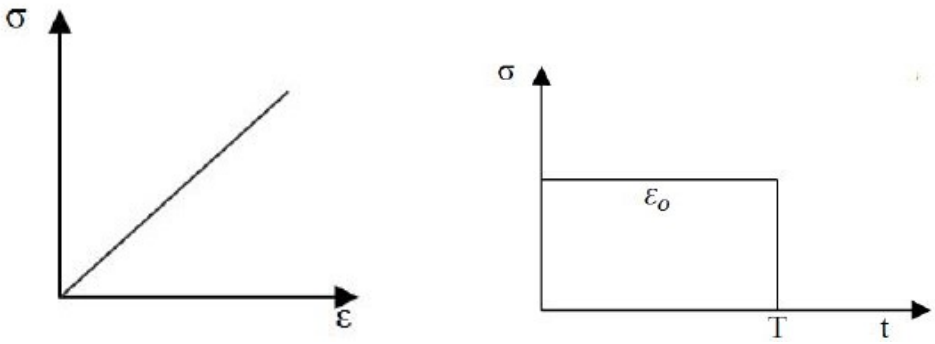


Figure 5.1: Spring to depict Linear Elastic behaviour

The relationship between stress and strain is represented by a straight line passing through the origin. Also, a plot between time and stress depicts the property of reversibility, where application of a constant strain makes the stress reach a constant value, and in the absence of any strain, the stress goes to zero. In order to derive the relation between the stresses in one direction and strains in other directions, tests such as CRS test, DSS test and Biaxial tests have been used.

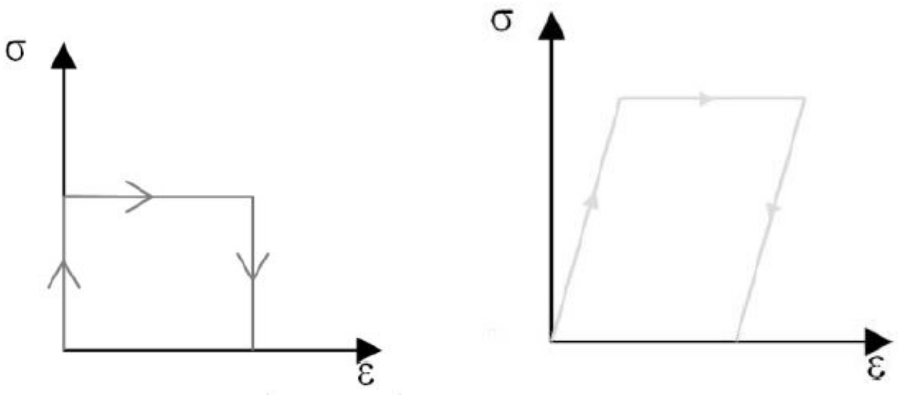


(a) Strain vs Stress (b) Time vs Stress

Figure 5.2: Linear Elastic soil behaviour

5.1.2 Mohr-Coulomb model

A Mohr-Coulomb soil model demonstrates a linearly elastic, perfectly plastic behavior and represents first order approximation of soil behavior; i.e., when the stress applied on the soil is below the yield stress, the soil has an elastic behavior. This was elaborated in detail in chapter 2. When a stress strain relationship for Mohr-Coulomb model (figure below) is plotted, it is observed that the influence of small stresses causes no deformations in the model. However, when the stress exceeds the yield stress limit, the soil undergoes plastic deformation. Until the stress is removed, the material remains at the constant stress level and the strain level is retained.



(a) Perfectly plastic (b) Linearly elastic perfectly plastic

Figure 5.3: Mohr-Coulomb soil behaviour

The aforementioned soil behavior can be explained through mechanics with the help of a spring and a slider. The spring demonstrates a linear behavior (following Hooke's law) and the plastic slider demonstrates the behavior of permanent deformation when applied upon a load higher than the plastic yield criterion. Combining both the mechanical elements in series, the behavior of linearly elastic and perfectly plastic material can be demonstrated.

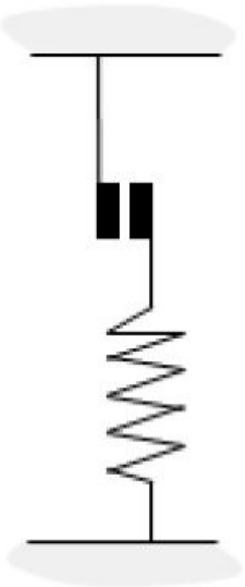


Figure 5.4: Arrangement to depict Mohr-Coulomb behaviour

5.2 Neural Network Implementation

Based on the behaviour of soil explained using mechanical components, the next step was the implementation of the Neural Networks for various soil behaviours using a similar method of implementation as the mechanical behaviour. Every typical behaviour like elasticity, plasticity and creep would be captured using neural networks and combined to simulate various complex soil behaviours.

5.2.1 Linear Elastic soil

As previously mentioned, the linear elastic soil is represented using a linear spring. In the neural networks model, this behaviour has been represented using a linear activation function. In order to fully capture the linear elastic behaviour of the soil, the relations between the strains and stresses in every direction were required. This has been made possible by taking into consideration various lab tests that gave out a relation between strains and stresses in each direction. A CRS test was used with a range of strain increments varying from $-5e-3$ to $2e-3$ in a normal distribution. Through this test (described in chapter 2), a relation between the strain in y-direction and stresses in x, y, z directions have been obtained. Using the same range of strain increments, a biaxial test or a plane-strain test has been carried out. From this test, the data of stresses in x, y, z directions has been obtained when a strain increment is applied in the x, z directions. Through this lab test therefore, a relation between the strain in x, z directions and stresses in x, y, z directions has been obtained. For shear directions, DSS tests with same range of strain increments have been used to derive the relation between the shear strain and shear stress. In order to train the neural network model, 4 laboratory tests of each soil test(consisting of 10000 data points each) have been used which were concatenated into a single dataset(total of $4*4*10000$ data points) for training and validation. The network has then been tested for prediction accuracy using one dataset of every soil test(one dataset from CRS, Biaxial, DSS tests consisting of 10000 datapoints each). This would provide with information in understanding the capability of the network as to where it is able/unable to predict the soil behaviour.

During the implementation of the Neural network for Linear-Elastic model, strain increment and stress increment have been considered as input and output respectively. This model consists of 6 neurons as input representing the strain increments in six directions, and 6 neurons representing the stress increments in these six directions. The linear relation is represented using a simple neural network as follows:

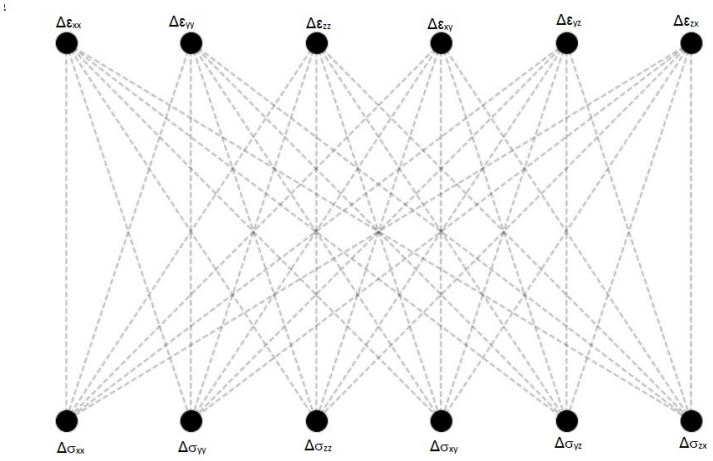


Figure 5.5: Neural network for linear elastic model

From the figure, it can be observed that every neuron of input is connected with every neuron of output. These connections have been represented by a 'Material Stiffness Matrix' in terms of weights which are a mathematical relation between the strain in a direction with the stress in the same or a different direction. Unlike, the neural network in chapter 4, the neural network in this chapter consists of single set of weights connecting the inputs and outputs. Therefore, equation (4.16) reduces to

$$\frac{\partial^{n+1}\Delta\sigma_i}{\partial^{n+1}\Delta\epsilon_j} = \frac{S_i^\sigma}{S_j^\epsilon}\sum_{k=1}^{N\sigma} w_{ij}^{\sigma\epsilon} \tag{5.1}$$

Based on equation (5.1), following were the theoretical and obtained values in the matrix :

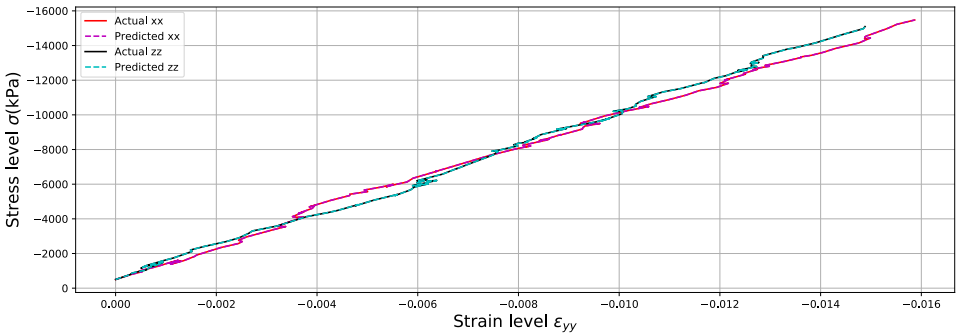
673076.92	288461.54	288461.54	0	0	0
288461.54	673076.92	288461.54	0	0	0
288461.54	288461.54	673076.92	0	0	0
0	0	0	192307.69	0	0
0	0	0	0	192307.69	0
0	0	0	0	0	192307.69

Table 5.1: Theoretical linear elastic matrix

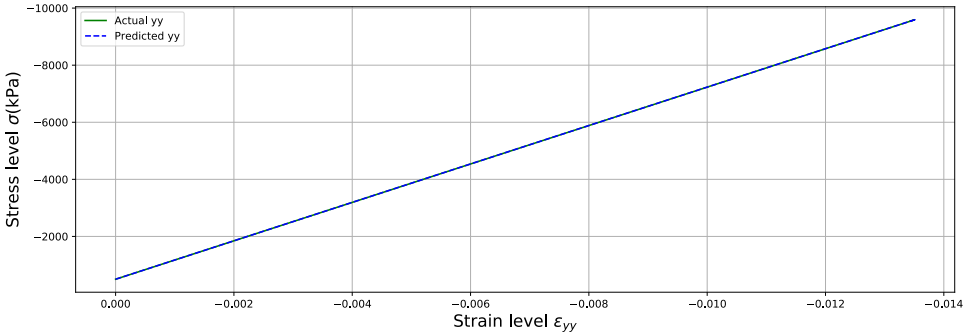
673038.5	288353.62	288431.28	0	0	0
288299.47	672851.25	288364.44	0	0	0
288387.97	288348.00	673072.25	0	0	0
0	0	0	192273.77	0	0
0	0	0	0	192408.95	0
0	0	0	0	0	192343.14

Table 5.2: Obtained linear elastic matrix

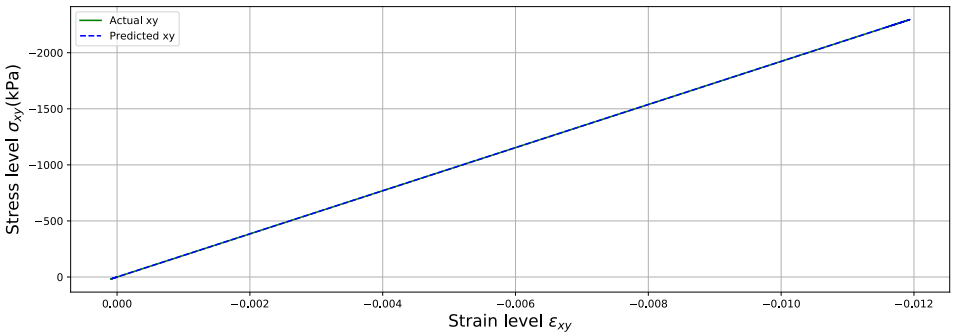
It can be seen that the error is significantly small and the matrix is close to being accurate. Following have been the results obtained for the prediction of the stress increments for given strain increments when the network was trained for 3 epochs with a batch size of 32.:



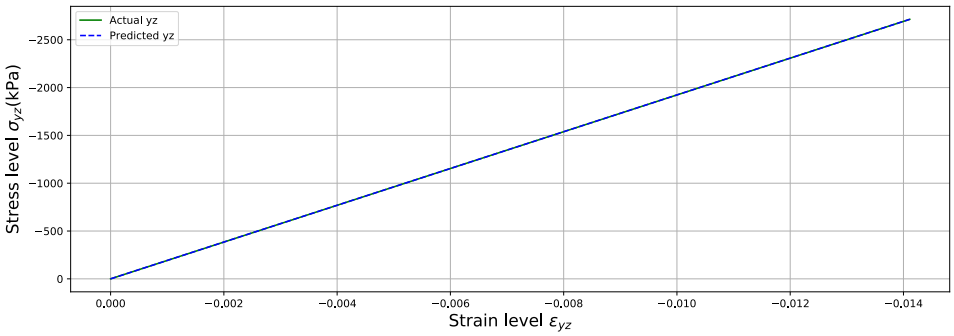
(a) Stresses in xx and zz directions in Biaxial test for testing dataset



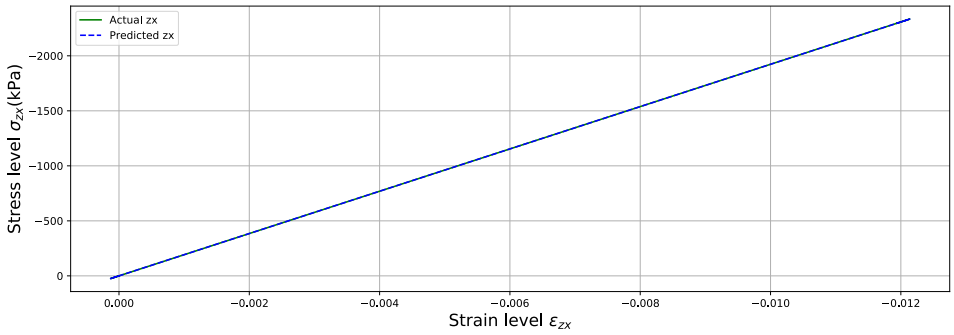
(b) Stresses in yy direction in CRS test for testing dataset



(c) Stresses in shear direction xy



(d) Stresses in shear direction yz for testing dataset



(e) Stresses in shear direction zx for testing dataset

Figure 5.6: Prediction for Linearly elastic soil

From figure 5.6 an inference has been made that, a linear behavior is obtained when the stresses in each direction are plotted against the strain in the y-direction. A similar behavior has been observed for the shear stresses plotted against shear strains. Further to analyze the accuracy of the model, the absolute error has been computed for each soil test dataset and the results are as follows:

Test	Error
Biaxial Test	0.02466
CRS Test	0.024786
DSS Test(xy)	0.02476
DSS Test(yz)	0.02476
DSS Test(zx)	0.024756

Table 5.3: Absolute error for prediction of Linear Elastic behaviour from Mohr-Coulomb ANN model

From the table above the maximum error observed is 0.024786 which can be considered insignificant. Therefore, it can be concluded that the neural network is accurate in predicting Linear elastic behaviour.

5.2.2 Mohr-Coulomb

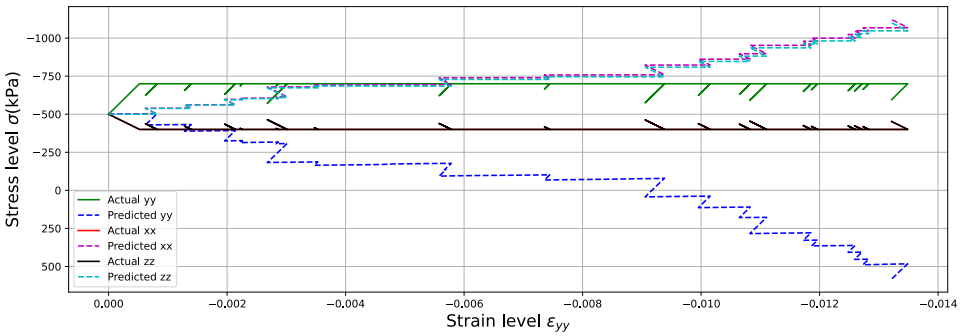
In the case of Mohr-Coulomb model, the objective was to predict the elastic and plastic behaviour of soil. As mentioned earlier, the plasticity occurs in the soil when the stress exceeds the failure criterion, i.e., soil demonstrates plastic behaviour at failure. Hence for the prediction of Mohr-Coulomb soil, data for the stresses in soil in the elastic and the plastic zone were required. The soil properties for this case were considered as follows:

$Property$	$Value$
$E(MPa)$	0.5
ν	0.3
ϕ	15.0°
$c(kN/m^2)$	8.0
ψ	0.0

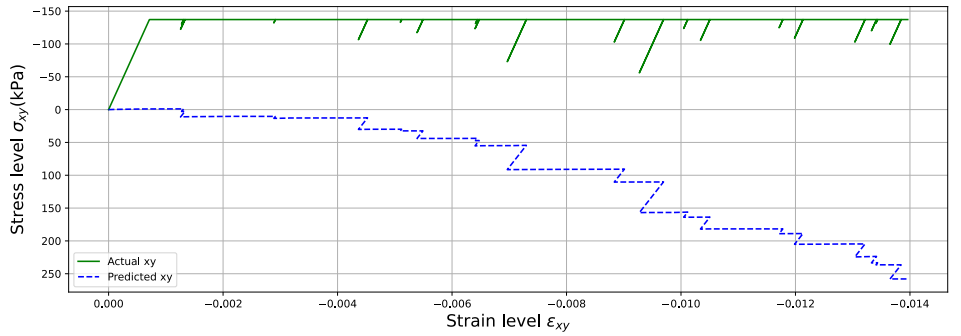
Table 5.4: Soil Properties for Mohr-Coulomb model

In the case of linear elastic model, CRS test has been used as a medium to obtain a relation between strain in y-direction and stresses in the principal directions. However, in CRS test, soil failure does not take place. This implies that only the elastic behavior of the Mohr-Coulomb could be captured through CRS test. In the case of a triaxial test, there is a possibility of soil failure. However, only the relation between the strain and stress in y-direction could be obtained from this test. Therefore, in order to capture the entire behavior of soil in elastic and plastic states and deduce a relation between the strains and stresses in the principal directions, a drained True Triaxial test has been used(Labuz and Zang, 2012). For the shear directions a drained DSS test has been used which was expected to give the relation between the shear strains and the shear stresses. The training datasets were generated for which random values of strain increments(y-direction) were selected randomly from the uniform distribution ranging from $-5e-4$ to $2e-4$. This specific range was selected to demonstrate the unloading-reloading behaviour of the soil. The strain increments in the xx and zz directions were half of strain increment in y-direction i.e., $\delta\epsilon_{xx} = \delta\epsilon_{zz} = -\delta\epsilon_{yy}/2$.

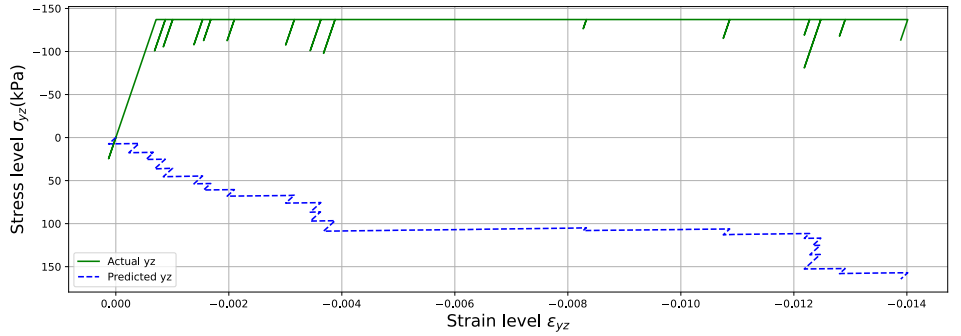
The mechanical model aforementioned(figure 5.4) has been used as an inspiration to model a neural network in order to predict the linearly-elastic, perfectly-plastic behavior. The linear behaviour and the plastic behaviour could be predicted using two ReLU activation functions. The model initially consisted of only the strain increments as input yielding stress increment as output. The model was constructed with two hidden layers of 24 neurons with ReLU activation functions(in both hidden layers). In order to avoid the predictions from diverging the data is normalized between the range of 0 and 1 using *MinMaxScaler*(Pedregosa et al., 2011). Using a trial and error method, the model(with the best predictions) has been trained for 40 epochs with a batch size of 1024. The results obtained were as follows:



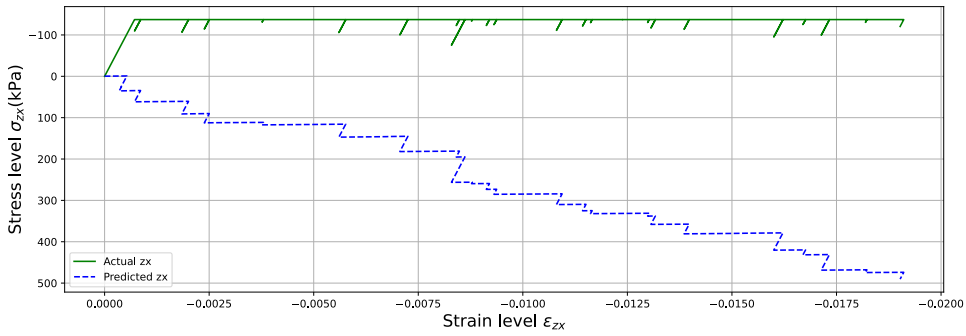
(a) Prediction of stress in principal directions for single input of strain increment for testing dataset of True triaxial test



(b) Prediction of stress in shear direction xy for single input of strain increment for testing dataset of DSS(xy) test



(c) Prediction of stress in shear direction yz for single input of strain increment for testing dataset of DSS(yz) test

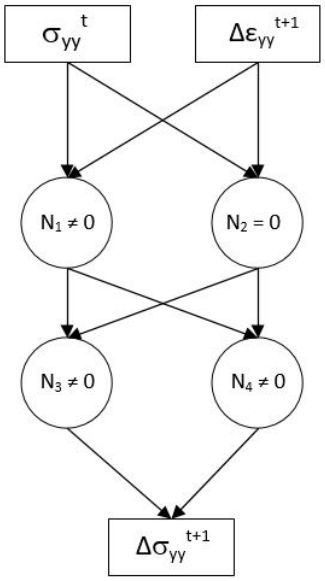


(d) Prediction of stress in shear direction zx for single input of strain increment for testing dataset of DSS(zx) test

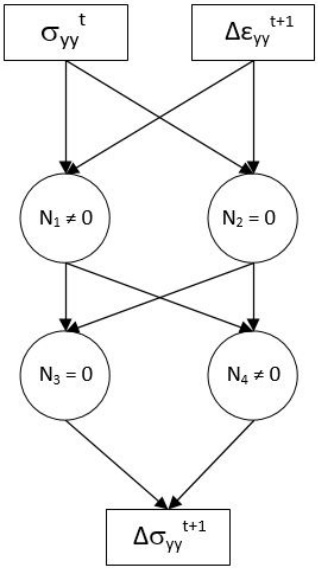
Figure 5.7: Prediction of Neural network model

It has been observed that the neural network model was unable to predict the soil behaviour and actually diverged. Therefore, with the increase in complexity in the model, a requirement of an additional input parameter was felt.

In order to improve the accuracy of the model, along with the strain increment, a second input of stress state has been considered. Historic data such as stress state help the network to consider the gradient of the mechanical response in its predictions(Ghaboussi and Sidarta, 1998a,Gulikers, 2018). The network was first modeled for y-direction to study its behavior in the case of elasticity and plasticity and to acquire the minimum number of neurons for the model for it to accurately predict. It has been observed that two neurons are required in each hidden layer to accurately predict the stress increments. The output of a neuron is the product of the input(to the neuron) and the weight added with the bias passing through an activation function. To understand the inner working of the neural network, the outputs of neurons have been plotted for an increment of every 100 values(in Appendix). As the prediction progressed the output of the neurons varied with change in state from elastic to plastic or plastic to elastic. It has been observed that during the elastic state, three neurons that is N_1, N_3, N_4 are activated in the network which help in predicting the stress increment. In the plastic state, only $N_3 = 0$, but N_1, N_4 are not equal to zero.



(a) Neurons during elasticity



(b) Neurons during plasticity

Figure 5.8: Outputs of neurons in elastic and plastic state

The model has been then extended to six directions progressively with the inclusion of inputs and neurons to predict the output(in figure below). The model has been constructed in such a manner that the strain-increment is an input to the entire network, but the stress state is given as an input in every direction individually(figure 5.9).

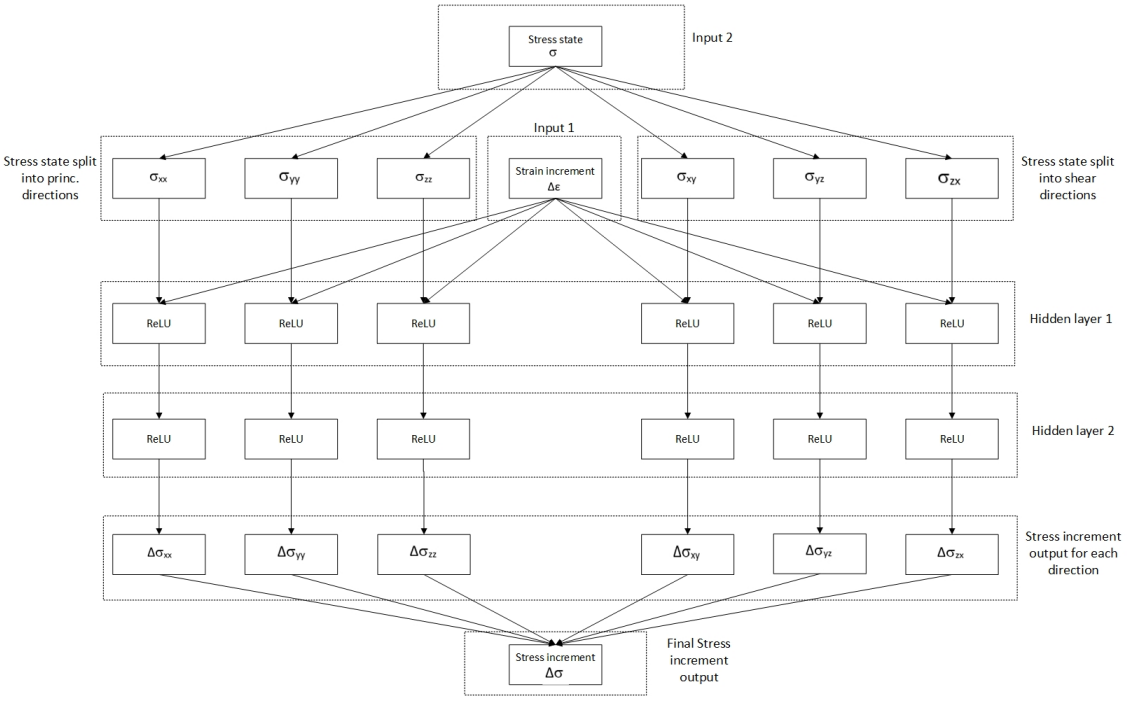


Figure 5.9: Neural network for Mohr-Coulomb model

With the generated data, the model has been trained with the data with a train-validation split of 20 percent, batch size of 256 and 40 epochs. For the training datasets, 9 True Triaxial and DSS(in each direction) artificially created soil tests were used. These soil tests were concatenated to create a single

dataset for training. The model was then tested for one soil test each(True triaxial test and DSS tests for every shear direction). Due to the increase in the complexity of the model, the number of tests(in comparison with Linear Elastic model) also had to be increased. During the testing of the model, the generated output of stress increment has been given as a feedback to the network which sums up with the previous stress state to generate a stress state for current time step. Therefore, the model requires an initial value of stress state to be given which has been set to be $-500kPa$ in the principal directions. For a better understanding, a flow chart of algorithm for testing the neural network is presented below:

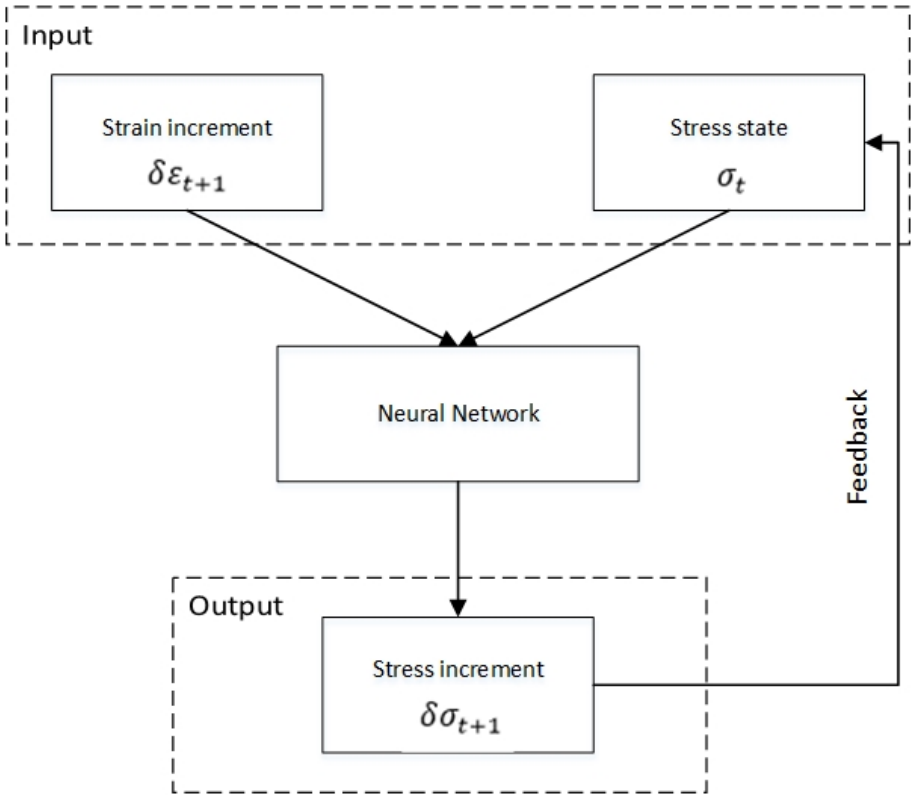
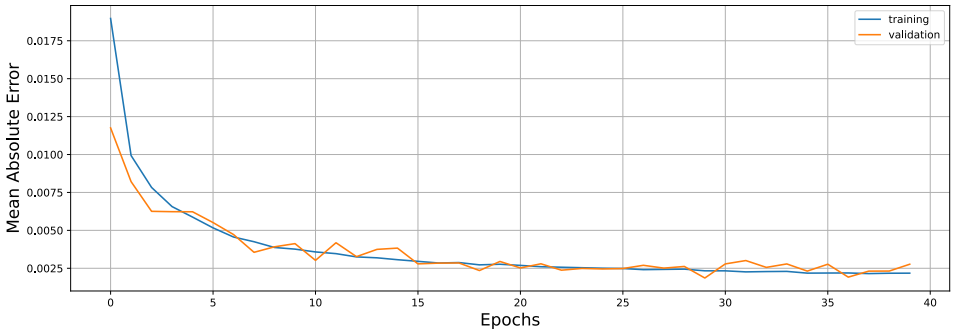
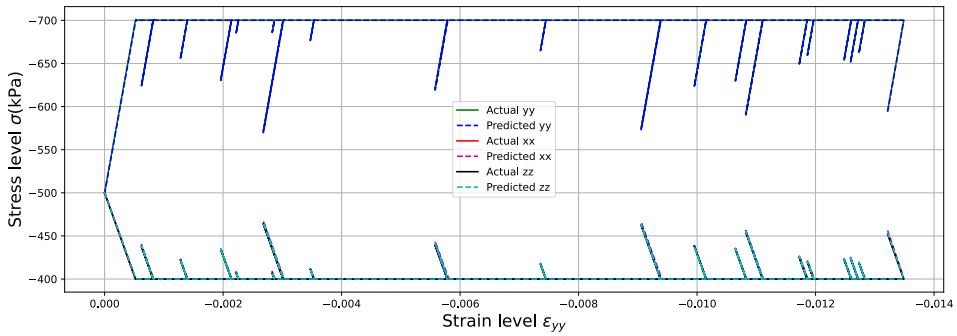


Figure 5.10: Flowchart for Neural network testing algorithm

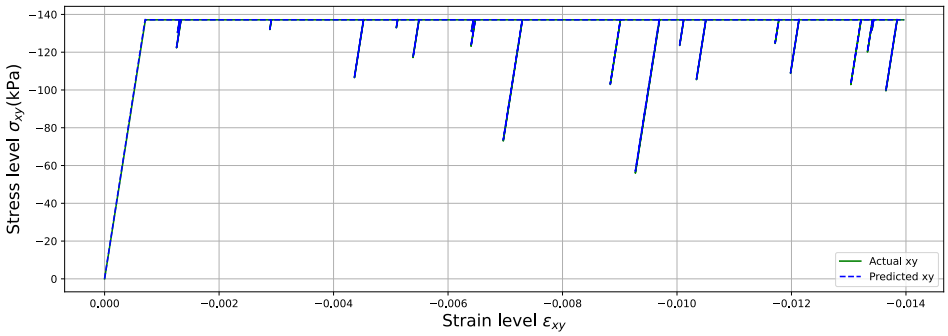
Following were the results obtained for the prediction of the stress increments for given strain increments and stress states:



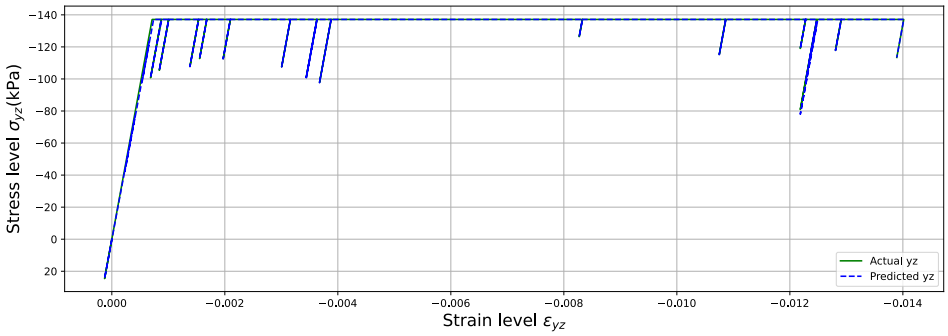
(a) Training and validation error vs Epochs



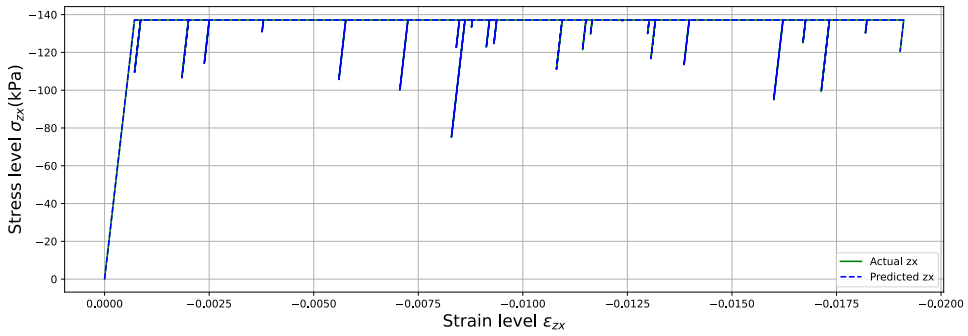
(b) Prediction of stress in principal directions for testing dataset



(c) Prediction of stress in shear direction xy for testing dataset



(d) Prediction of stress in shear direction yz for testing dataset



(e) Prediction of stress in shear direction zx for testing dataset

Figure 5.11: Prediction of Neural network model for testing datasets

It is observed that the neural network has been able to predict the Mohr-Coulomb behavior accurately. The mean absolute errors of the model are as follows:

Test	Mean Absolute Error
True Triaxial	$2.4838 * 10^{-2}$
Direct Shear(xy)	$2.6709 * 10^{-2}$
Direct Shear(yz)	$2.7530 * 10^{-2}$
Direct Shear(zx)	$2.8202 * 10^{-2}$

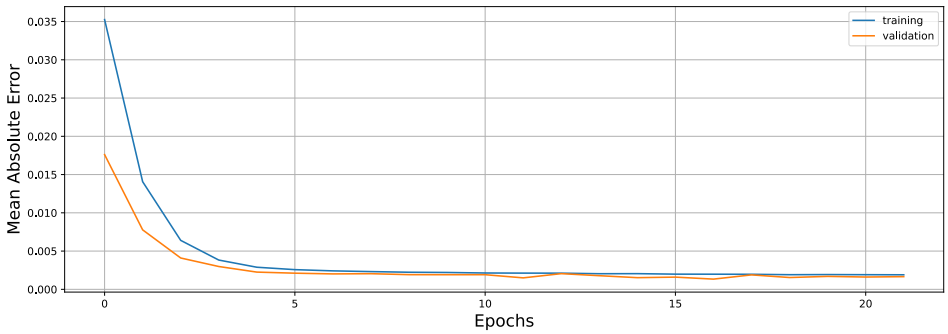
Table 5.5: Absolute error(in kPa) for the neural network model during testing

From this it can be inferred that the accuracy of the model in predicting the required values is very high, but the model is not able to learn the behavior of zero output for a given zero input in the model. The output error for zero input would gradually accumulate to produce a final result with a significant error. This has mainly been observed in the case of DSS tests. Therefore, the number of datasets for DSS tests have been increased by 1. In order to avoid over-training and to improve the generalization ability of the model, a dropout layer was introduced in the model. To avoid further overfitting of the model, early-stopping was applied in the model with a patience of 5 and the model was trained for 40 epochs. The value for the dropout was determined by running a search algorithm with various values of dropout such as 0.20, 0.15, 0.10 and 0.05. The plots are presented in the Appendix and absolute errors for every dropout are represented in the table below:

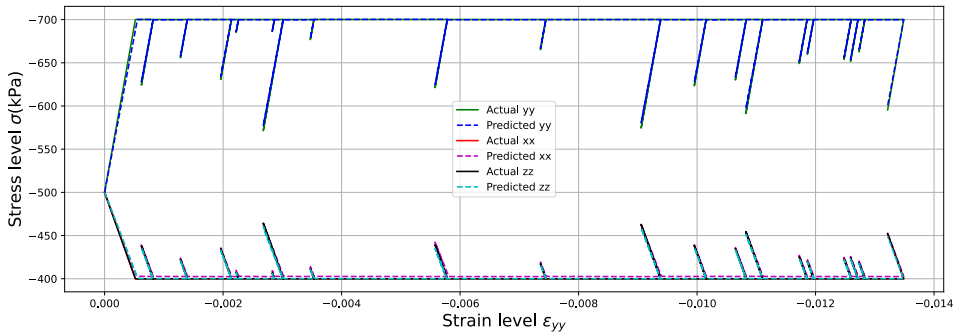
Dropout	Absolute Error					
	$\delta\sigma_{xx}$	$\delta\sigma_{yy}$	$\delta\sigma_{zz}$	$\delta\sigma_{xy}$	$\delta\sigma_{yz}$	$\delta\sigma_{zx}$
0.20	$4.98 * 10^{-2}$	$4.93 * 10^{-2}$	$4.94 * 10^{-2}$	$1.49 * 10^{-2}$	$0.85 * 10^{-2}$	$0.24 * 10^{-2}$
0.15	$4.96 * 10^{-2}$	$4.98 * 10^{-2}$	$4.89 * 10^{-2}$	$0.011 * 10^{-2}$	$0.01 * 10^{-2}$	$0.02 * 10^{-2}$
0.10	$4.98 * 10^{-2}$	$4.98 * 10^{-2}$	$4.934 * 10^{-2}$	$0.015 * 10^{-2}$	$0.009 * 10^{-2}$	$0.001 * 10^{-2}$
0.05	$4.94 * 10^{-2}$	$4.99 * 10^{-2}$	$4.98 * 10^{-2}$	$0.0033 * 10^{-2}$	$0.002 * 10^{-2}$	$0.006 * 10^{-2}$
0.00	$4.95 * 10^{-2}$	$4.93 * 10^{-2}$	$4.96 * 10^{-2}$	$0.02 * 10^{-2}$	$0.002 * 10^{-2}$	$0.007 * 10^{-2}$

Table 5.6: Absolute error(in kPa) for various dropout values

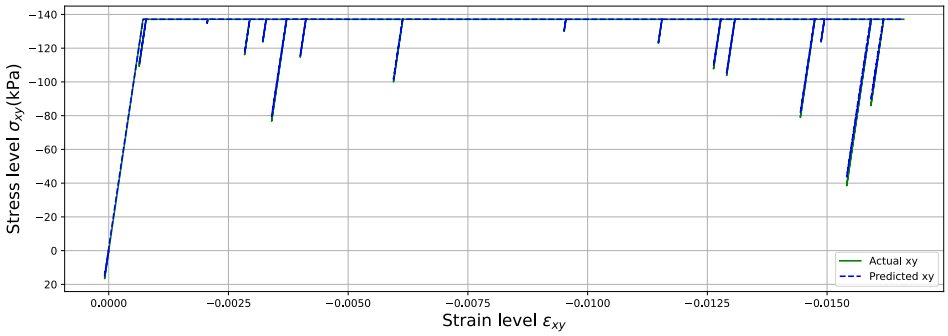
From the results of various dropouts and the obtained plots(in Appendix fig.A.2), an apt neural network model for the prediction of Mohr-Coulomb behaviour has been determined with a dropout of 0.05 and a batch size of 128. The prediction of the model with respect to the theoretical values are as follows:



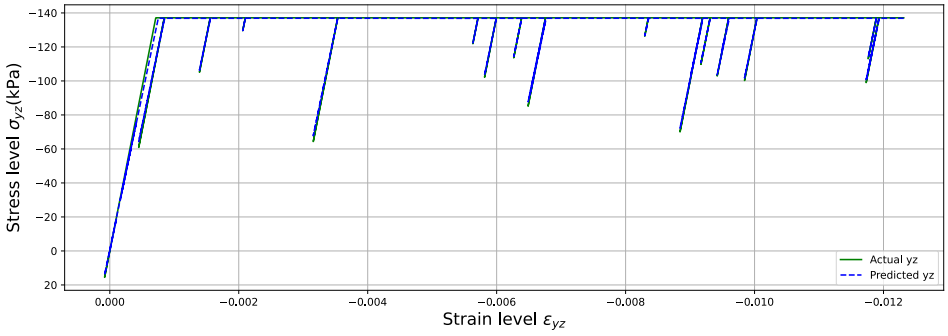
(a) Training and validation error vs Epochs for improved network



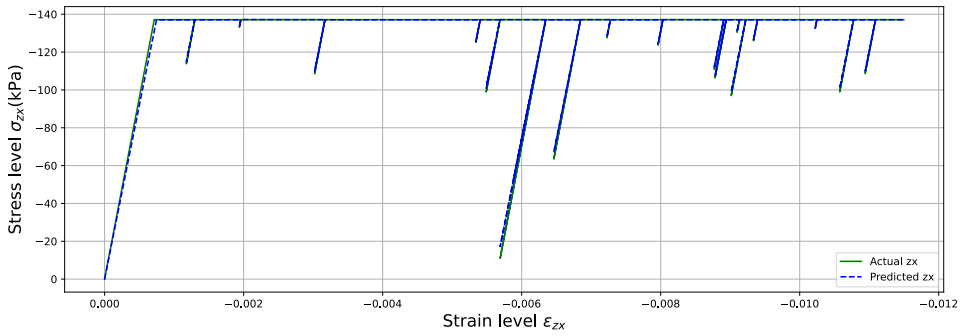
(b) Prediction of stress in principal directions for testing dataset for improved network



(c) Prediction of stress in shear direction xy for testing dataset for improved network



(d) Prediction of stress in shear direction yz for testing dataset for improved network



(e) Prediction of stress in shear direction zx for testing dataset for improved network

Figure 5.12: Prediction of the improved Neural network model

The devised neural network has been able to predict the Mohr-coulomb behaviour with a mean absolute error of 0.024. In figure 5.12b, it can be observed the model has been unable to accurately predict the plastic stress state. A similar behaviour can be observed in figure 5.12e where the model is unable to reach the stress value in unloading state. This inaccurate behaviour is due to the minor errors in the model during the prediction of stress increment which accumulate with time. All in all, the model despite consisting of errors, is able to predict elasto-plastic behaviour with a decent accuracy.

5.2.3 Generalization ability of model

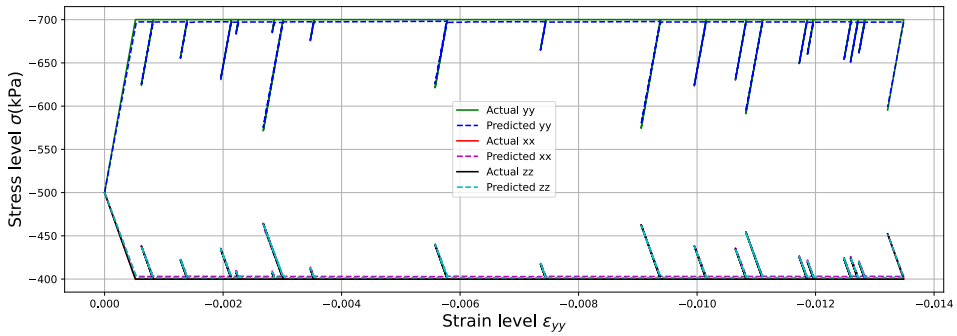
The objective of the neural network model is to be applicable in a real-life test scenario. In a real-life test, the data might contain noise which the model must be able to eliminate to give accurate predictions. Hence the generalization ability of the model has been investigated to check the limit of noise the model can withstand in order to provide accurate results. Two test cases have been considered to analyze the generalization ability of the model:

- Case 1: When the model is trained with noisy data
- Case 2: When the model is tested with a different initial stresses

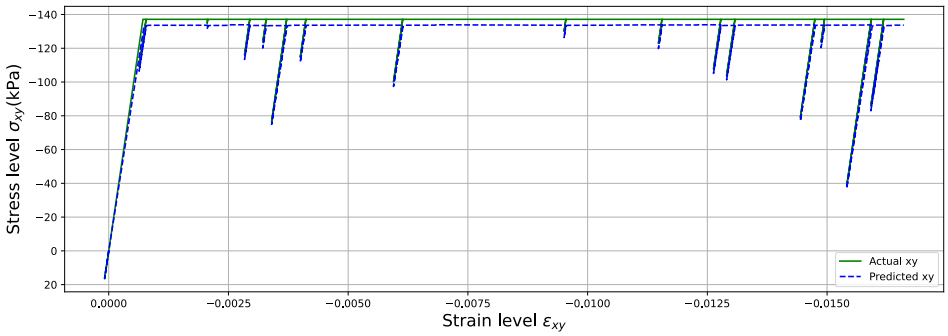
In the following sections, both the cases are elaborated in detail.

Noise in Training data

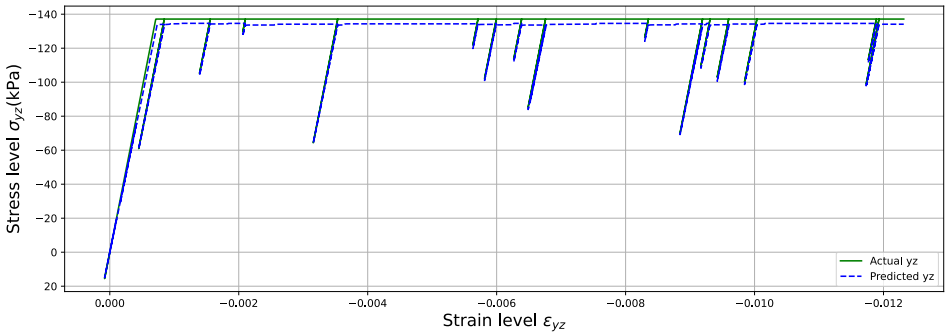
To simulate a real-life scenario, noise has been introduced in the stress state of the datasets using a random normal distribution. The training datasets are modified in such a way that 80% of the data consists of noise at random locations in the data. A sample of training dataset (for one artificial test)with noise has been presented in Appendix A. The final training dataset for the neural network would be a concatenation of such datasets. The model, then has been tested for a dataset with no noise to verify the generalization ability of the model. Various noisy datasets have been generated by changing the standard deviation in the normal distribution, with a constant mean of 0. The trained model was tested with the existing datasets. The training datasets were initially induced a noise with a standard deviation of $2kPa$. The results for the network were as follows:



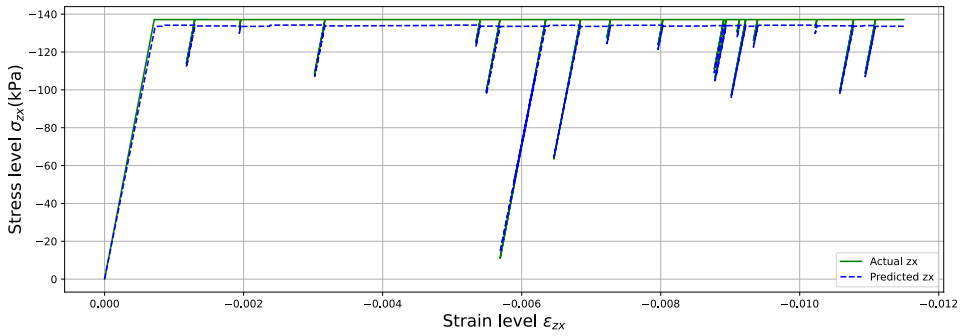
(a) Prediction of stress in principal directions for testing dataset for noise of 6kPa



(b) Prediction of stress in shear direction xy for testing dataset for noise of 6kPa



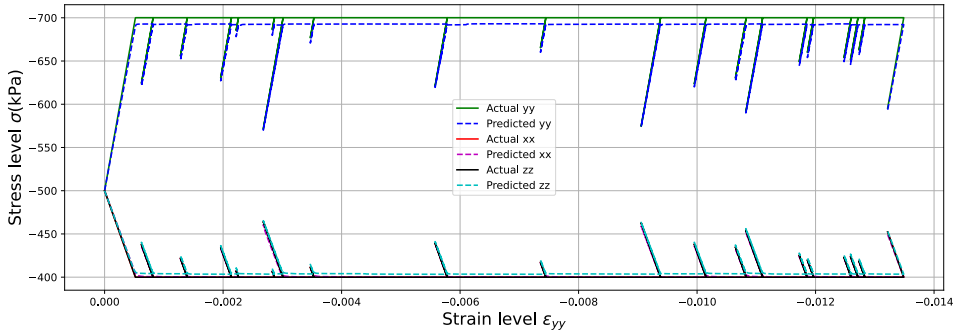
(c) Prediction of stress in shear direction yz for testing dataset for noise of 6kPa



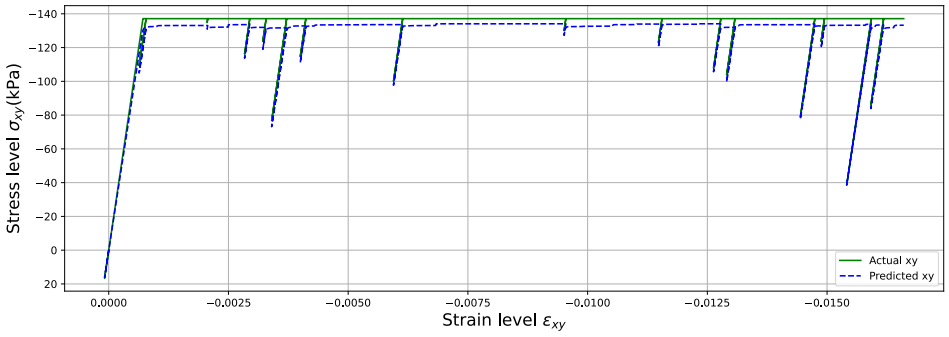
(d) Prediction of stress in shear direction zx for testing dataset for noise of 6kPa

Figure 5.13: Prediction of Neural network model trained with noise with standard deviation $2.0kPa$ or $\pm 6kPa$

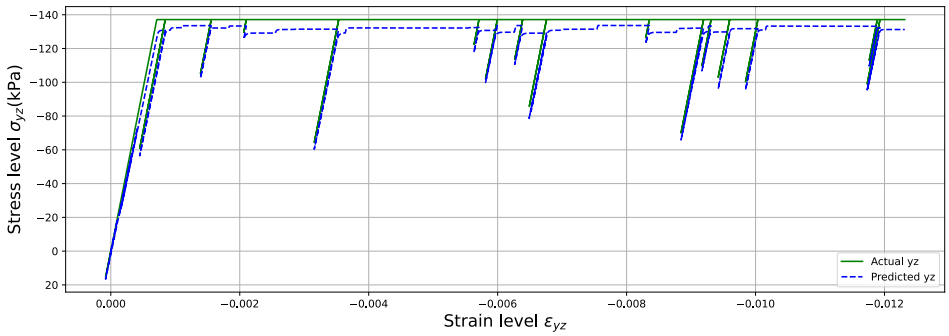
From the above plots, it can be inferred that for a noise of $6kPa$(or standard deviation of $2kPa$), the model has been able to produce nearly accurate results. In the DSS test results, it can be observed that, due to the training of network with a noisy data, the network faces a minor issue in predicting the exact stress in plastic state. However, the error observed was $3kPa$ which can be considered not-very-significant. The accuracy of the model was then tested for a noise of $5kPa$(standard deviation) and the results were noted as follows:



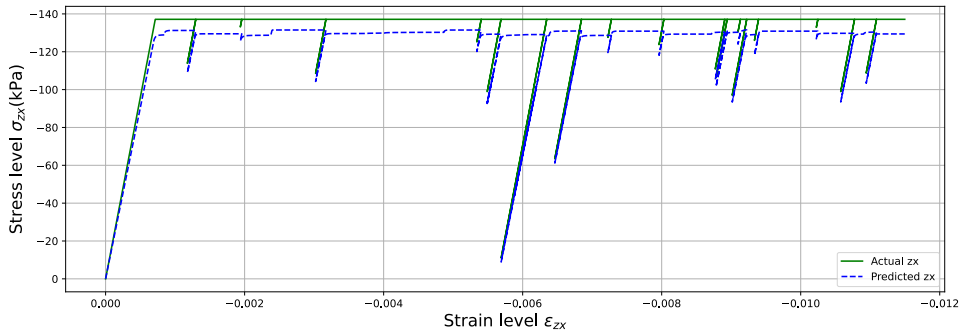
(a) Prediction of stress in principal directions for testing dataset for noise of 15kPa



(b) Prediction of stress in shear direction xy for testing dataset for noise of 15kPa



(c) Prediction of stress in shear direction yz for testing dataset for noise of 15kPa



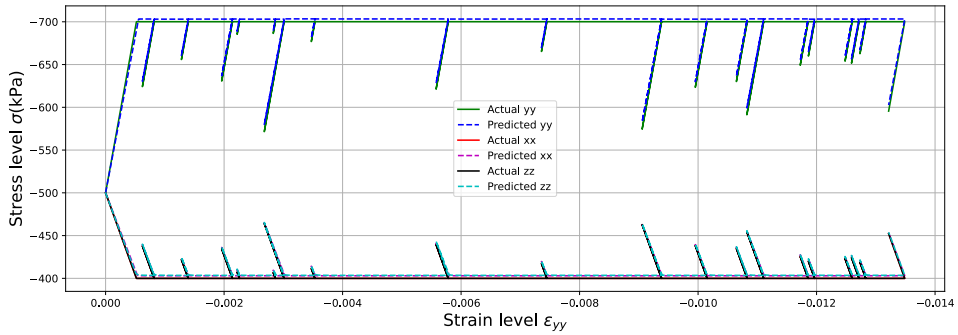
(d) Prediction of stress in shear direction zx for testing dataset for noise of 15kPa

Figure 5.14: Prediction of Neural network model trained with noise with standard deviation $5.0kPa$ or $\pm 15kPa$

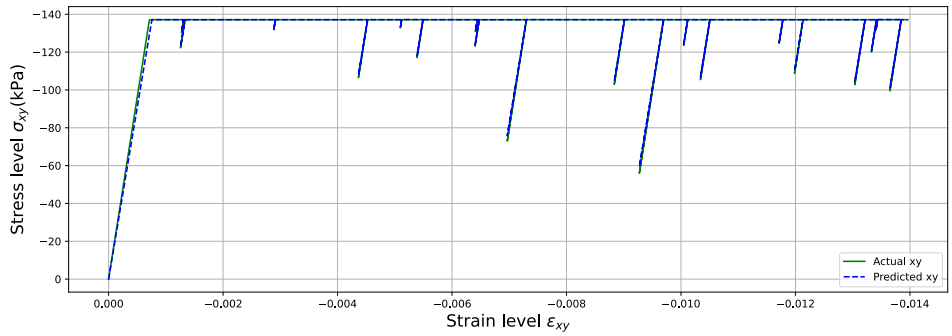
Based on the above plots, it can be inferred that the predictions are not very accurate and major deviations can be observed in the model in the shear tests. Though the model was not able to predict the values with an accuracy, it was observed that the network was able to predict the unloading-reloading trend of elasto-plastic soil.

Varying initial stresses in Testing data

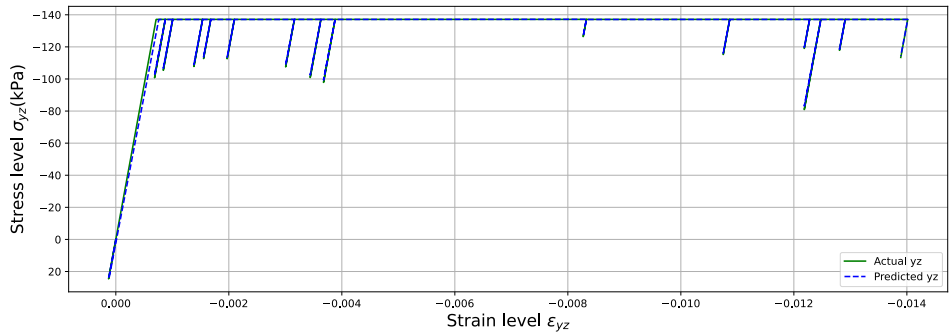
To further evaluate the generalization ability of the neural network, the network was trained with no noise in the data. However, in the testing phase, the initial stresses given to the model were changed to observe the accuracy of predictions. For the first case the initial stresses were set to be $-497kPa$ for which the results were as follows:



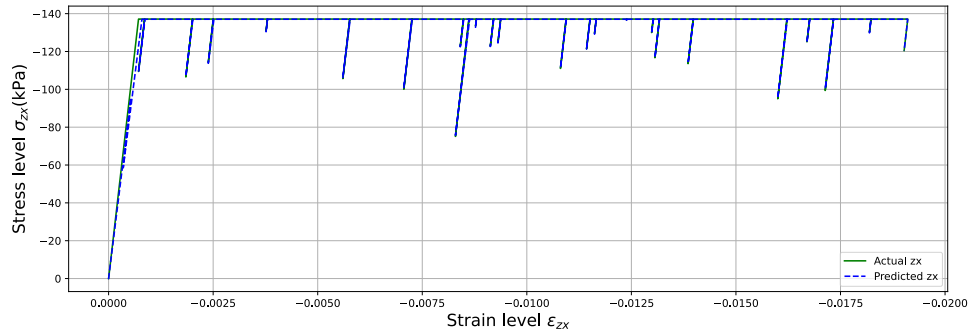
(a) Prediction of stress in principal directions for testing dataset for initial stress of $-497kPa$



(b) Prediction of stress in shear direction xy for testing dataset for initial stress of $-497kPa$



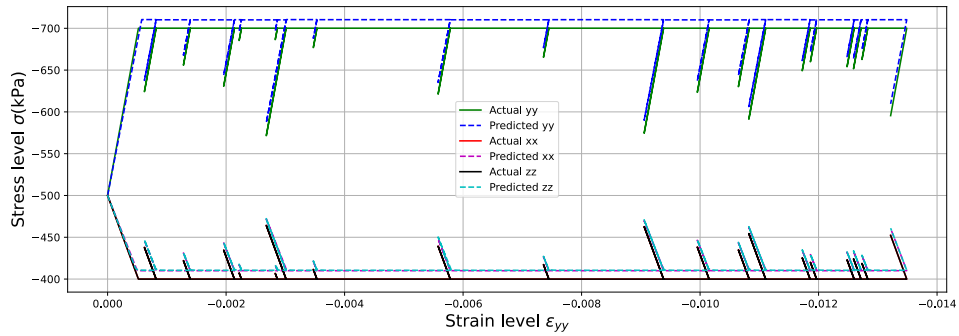
(c) Prediction of stress in shear direction yz for testing dataset for initial stress of $-497kPa$



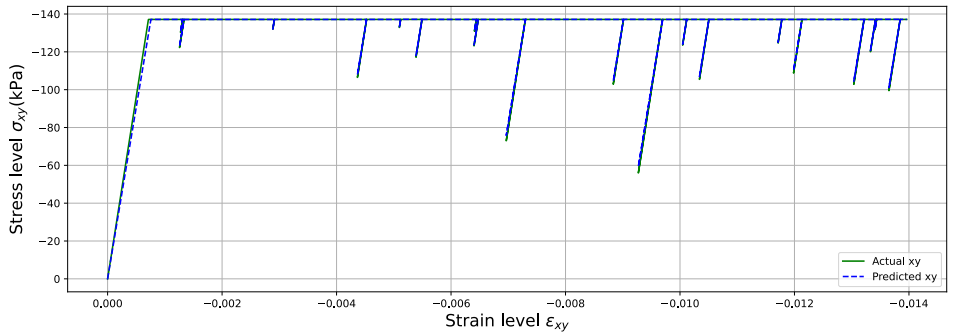
(d) Prediction of stress in shear direction zx for testing dataset for initial stress of $-497kPa$

Figure 5.15: Prediction of Neural network model tested for stress initialized to $-497.0kPa$

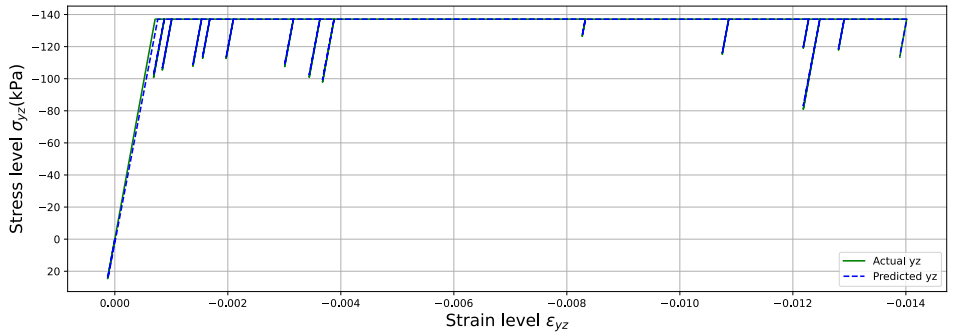
It has been observed that by changing the initial stresses to $-497kPa$, the model is able to predict the behavior with a significant accuracy. Since, the initial stresses are only applied in the principal directions, it can be inferred from the plots that, change in initial stresses has no effect on the prediction of the stresses in shear directions. The model is further tested for an initial stress of $-490kPa$. The results in this case were as follows:



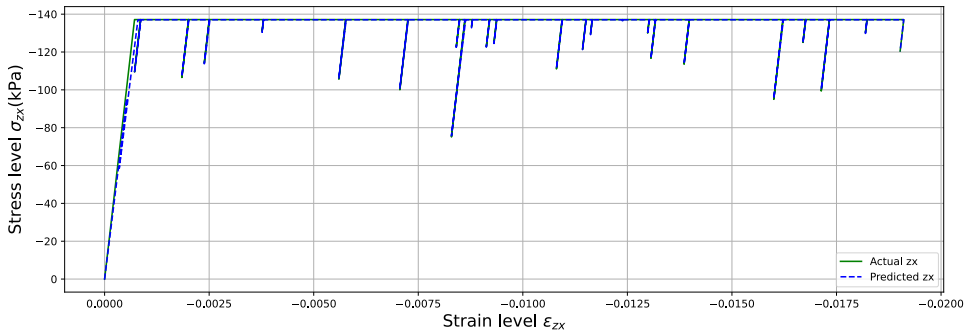
(a) Prediction of stress in principal directions for testing dataset for initial stress of $-490kPa$



(b) Prediction of stress in shear direction xy for testing dataset for initial stress of $-490kPa$



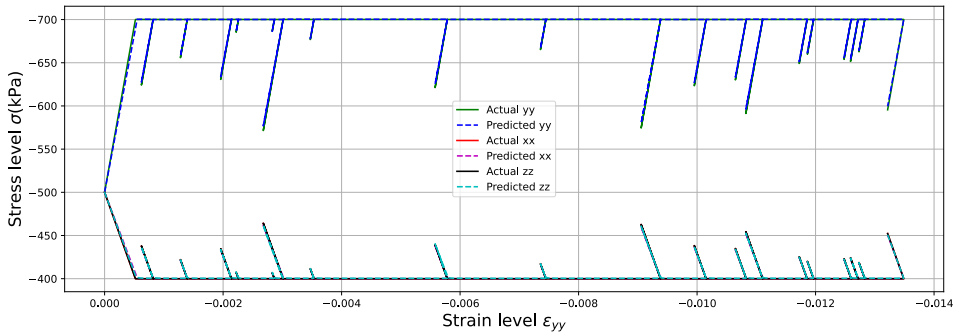
(c) Prediction of stress in shear direction yz for testing dataset for initial stress of $-490kPa$



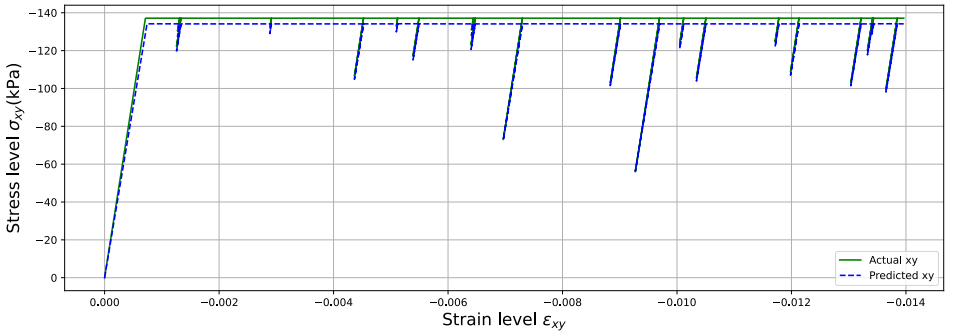
(d) Prediction of stress in shear direction zx for testing dataset for initial stress of $-490kPa$

Figure 5.16: Prediction of Neural network model tested for stress initialized to $-490.0kPa$

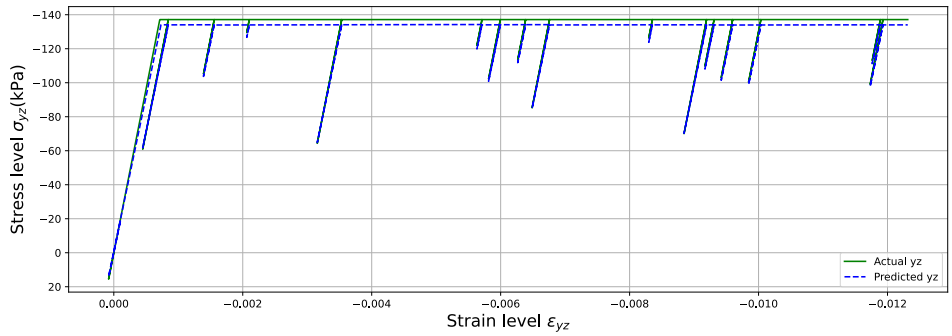
It has been observed from the prediction in principal directions, that an offset of values is taking place when reaching the plastic state. This error is due to the initial stresses which the model is unable to rectify. The prediction in shear directions are the same as the previous case. The model has further been checked for its generalization ability by inducing a deviation of $-3kPa$ in initial stresses in shear directions with the stresses in principal directions at $-500kPa$. The results of the predictions are as follows:



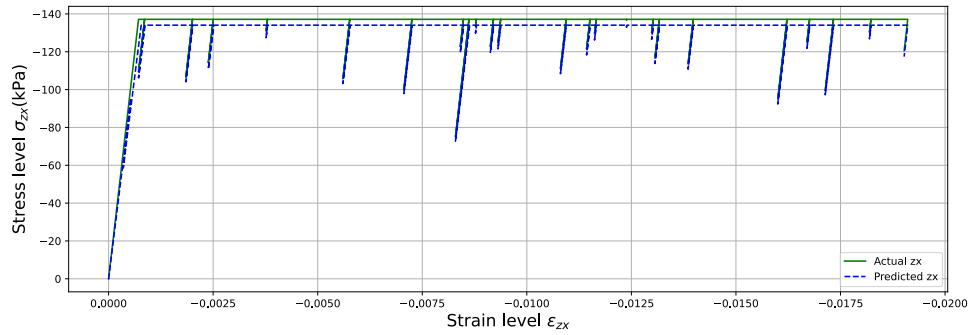
(a) Prediction of stress in principal directions for testing dataset for initial stress of $-3kPa$ in shear directions



(b) Prediction of stress in shear direction xy for testing dataset for initial stress of $-3kPa$ in shear directions



(c) Prediction of stress in shear direction yz for testing dataset for initial stress of $-3kPa$ in shear directions



(d) Prediction of stress in shear direction zx for testing dataset for initial stress of $-3kPa$ in shear directions

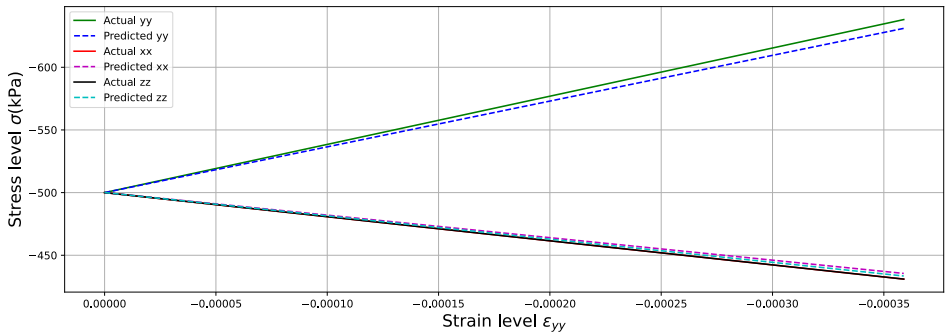
Figure 5.17: Prediction of Neural network model tested for stress initialized to $-3kPa$ in shear directions

The results observed in this case are having an opposite effect compared to the previous case of stress initialization. The predictions in the principal directions have been accurate, but the predictions in the shear directions have an offset. However, the model was able to correct the deviation and obtain a significant accuracy for prediction. The model was then tested for a stress initialization of $6kPa$(results in Appendix A.5). At $6kPa$, the model has not been able to predict the stresses with a great accuracy due to a high deviation from the required initial stress of $0kPa$. The absolute errors for various stress initialization is given in table A.1 in the Appendix.

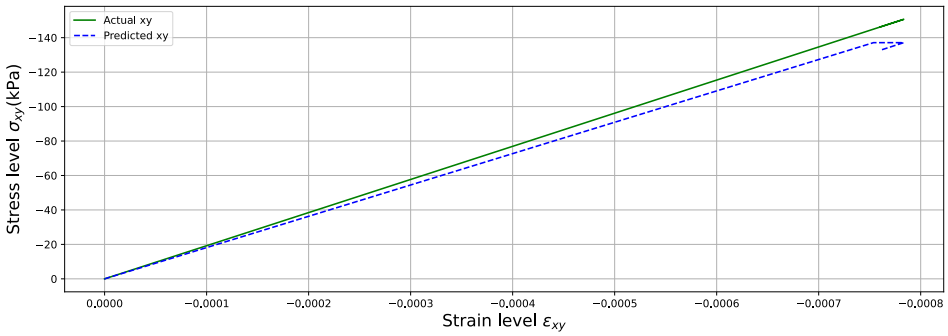
Practical Application of the model: In terms of the practical applicability, considering the training data extracted from a real-life lab test, a noise in the training data upto $\pm 6kPa$, would not affect the model significantly and the results would be nearly accurate. In terms of testing data, the In terms of testing data, the model will have a significant amount of accuracy for a deviation in initial stresses upto $3kPa$, beyond which the model may tend to have inaccuracies in the result.

5.2.4 Predicting simpler models using complex models

In theory, Mohr-Coulomb model demonstrates both elastic and plastic behaviour. Hence it has an ability to predict the linear elastic behaviour. Since, the neural network was trained with data of a Mohr-Coulomb model, a hypothesis was made that it would be able to predict the Linear Elastic behaviour as well. Therefore, a dataset for True Triaxial test and DSS-xy test were created using the Soil test option in PLAXIS 3D, considering the same parameters of minimum and maximum strain increments. Since, the model is Linear Elastic, it consists of no plastic limit, implying, there is a chance that with a introduction of higher number of phases, stresses could increase beyond the value which the Mohr-Coulomb model had never trained for and subsequently leading to divergence in predictions. Therefore, the number of phases are selected to be 4 which would generate stresses not higher than the plastic limit of the Mohr-Coulomb datasets. The results for the predictions were as follows:



(a) Prediction of stress in principal directions for Linear elastic model from Mohr-Coulomb ANN



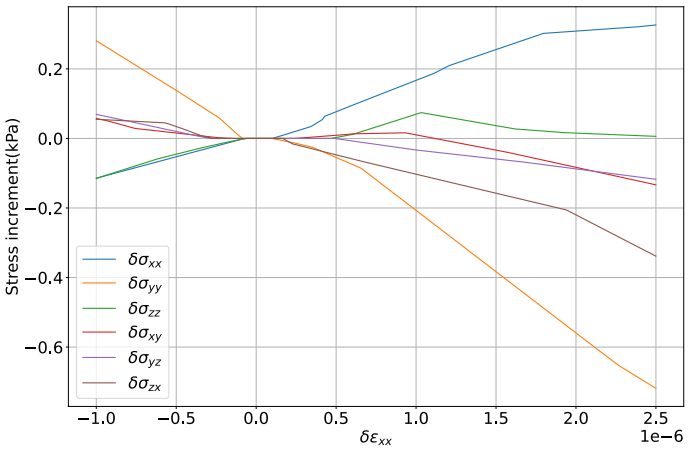
(b) Prediction of stress in shear direction xy for Linear elastic model from Mohr-Coulomb ANN

Figure 5.18: Prediction of linear elastic behaviour through Mohr-Coulomb ANN model

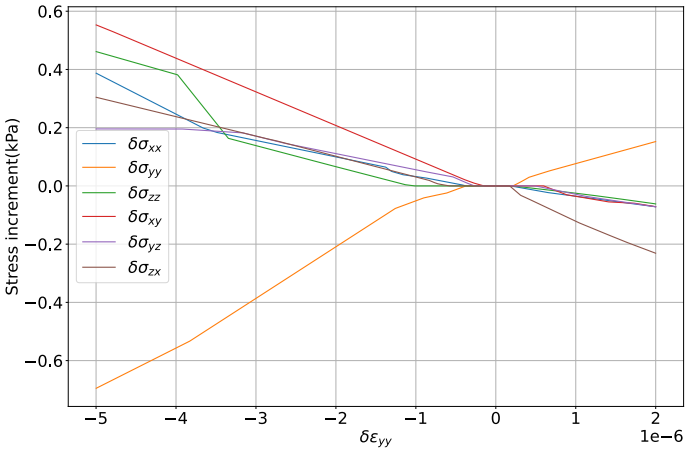
From the predictions above, it is evident that the neural network model has the ability to predict linear elastic behavior. The Mohr-Coulomb ANN model has been trained to reach a point of plasticity at $-137kPa$ after which unloading takes place. A similar behavior can be observed from figure 5.18b where the model at approximately $-137kPa$ reaches a point of plasticity followed by an unloading behavior.

5.2.5 Sensitivity Analysis

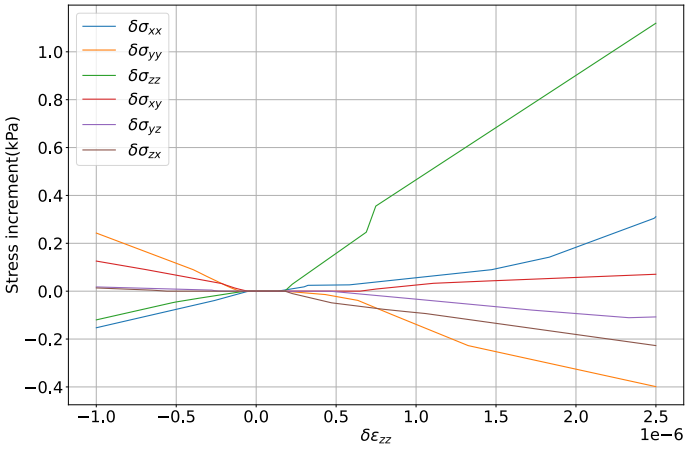
In this section, a sensitivity analysis of the inputs has been conducted to understand the relation between the network and the input variables. For this purpose, two cases are considered:
Case 1: In this, a sensitivity analysis of strain increment is demonstrated by assigning initial state as a fixed value to the stress states i.e., the principal directions have a stress of $-500kPa$ and the shear directions have zero stress. An increment is then induced to the strain increment input in each direction to evaluate the effect of strain increment on stress increment in terms of physical meaning. The minimum and maximum values to the strain increment are the minimum and maximum values on which the variable is scaled for the purpose of training.



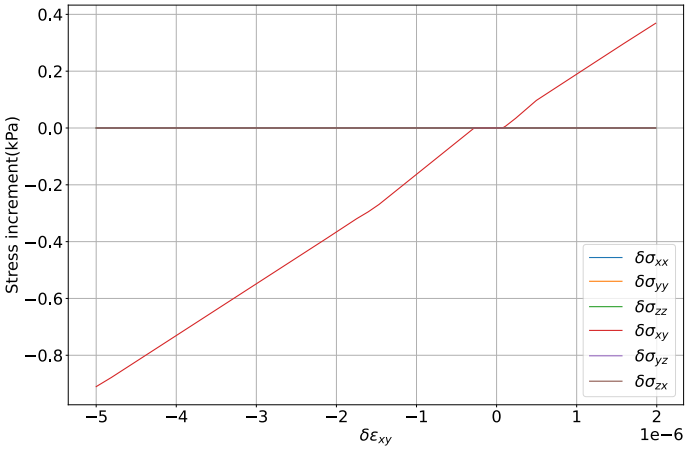
(a) Perturbation in $\Delta\epsilon_{xx}$



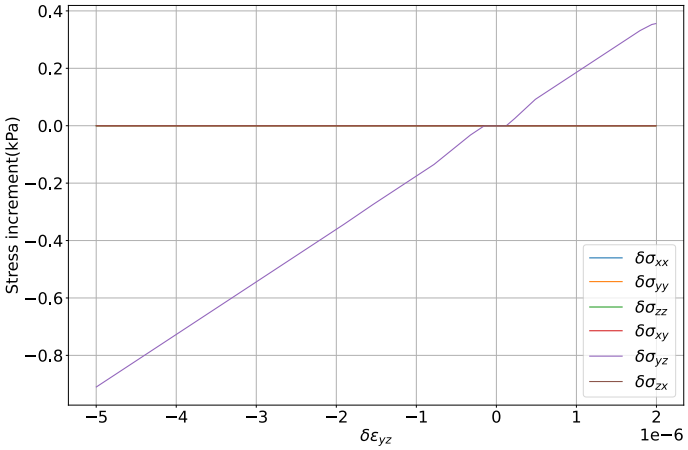
(b) Perturbation in $\Delta\epsilon_{yy}$



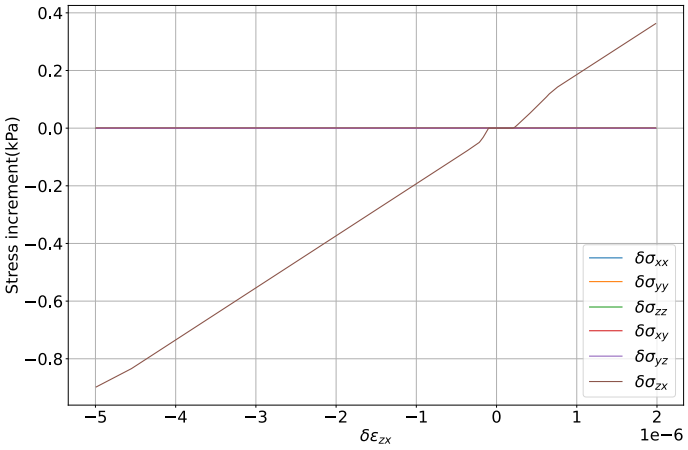
(c) Perturbation in $\Delta\epsilon_{zz}$



(d) Perturbation in $\Delta\epsilon_{xy}$



(e) Perturbation in $\Delta\epsilon_{yz}$

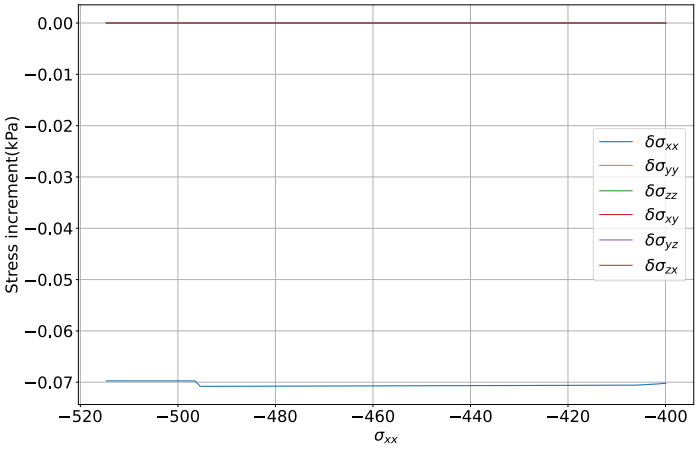


(f) Perturbation in $\Delta\epsilon_{zx}$

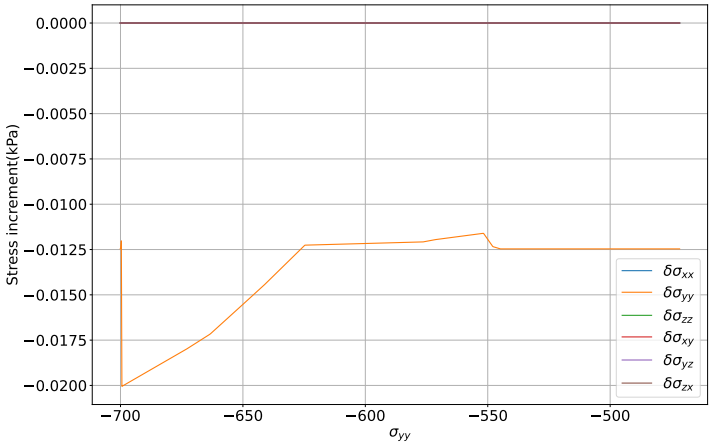
Figure 5.19: Sensitivity analysis for strain increment

It can be observed that a change in strain increment in x-direction leads to a similar amount of increase in stress increment in x-direction and z-direction(low increase is due to the error in the network). But in the y-direction the magnitude of increase is almost twice of stress increment in x-direction but in the negative direction. This is due to the model having been trained with datasets which consisted of similar information where the stress increment in y-axis is twice the magnitude and negative in direction with respect to x-axis and stress increment in z-axis is equal to the stress increment in x-axis. A similar trend can be observed in figures (5.19a) and (5.19b). The perturbation in the stress increments shear directions during the change in principal strain increments can only be attributed to the factor of error in the network. In figures (5.19d), (5.19e), (5.19f) where the strain increment is applied in the shear directions, it is observed that the shear increment in a certain direction leads to a shear stress in the same direction only and does not affect any other directions thus corroborating the DSS tests used to train the model. There is a region present where the shear stress increment is zero in the figures (5.19d), (5.19e), (5.19f). In this region, the amount of strain increment applied has a negligible effect on the output.

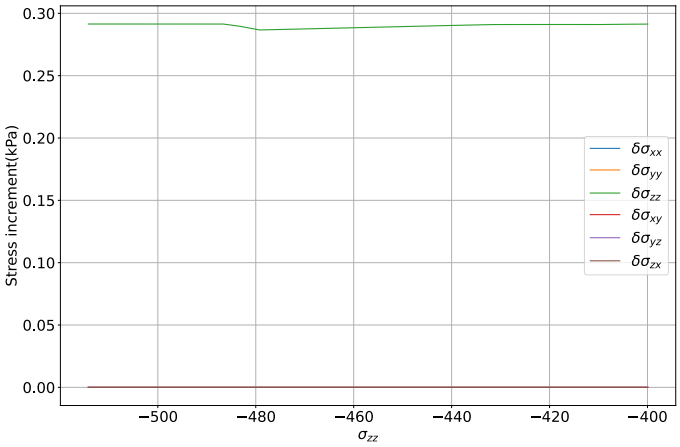
Case 2: The sensitivity of stress state is evaluated in this case. The input value of stress state is varied in each direction with the strain increment fixed at zero and the output is observed.



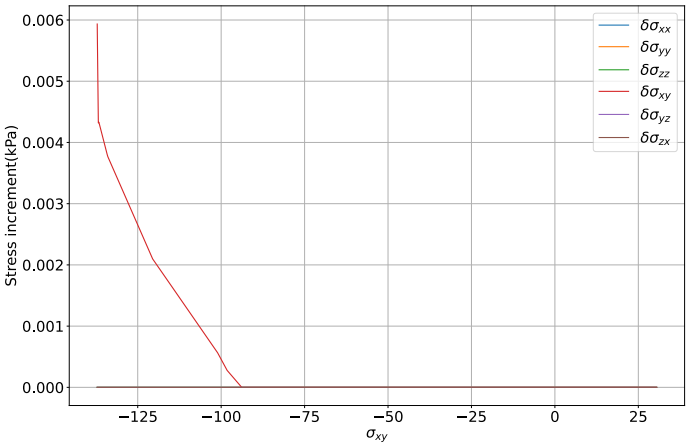
(a) Perturbation in σ_{xx}



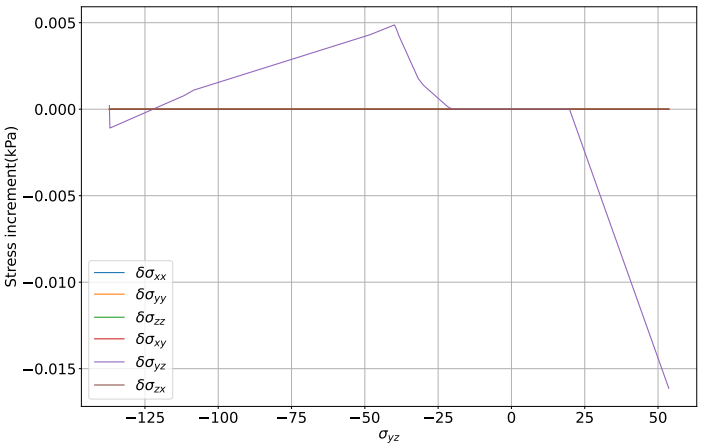
(b) Perturbation in σ_{yy}



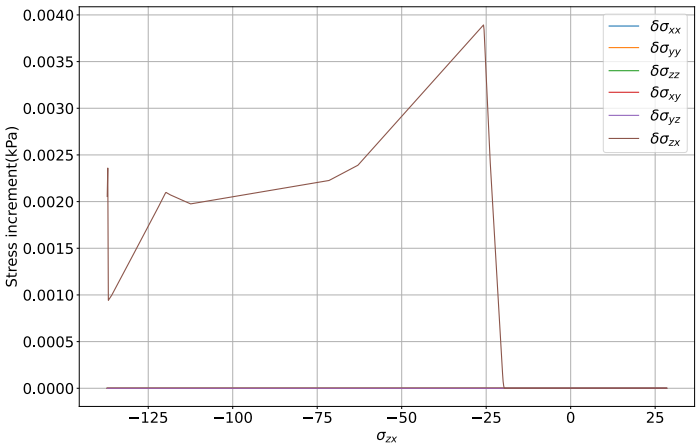
(c) Perturbation in σ_{zz}



(d) Perturbation in σ_{xy}



(e) Perturbation in σ_{yz}



(f) Perturbation in σ_{zx}

Figure 5.20: Sensitivity analysis for stress state

There is no particular physical relation between the stress increment and stress state except that the new stress state is the summation of the previous stress states and the current stress increment. The behavior of the stress increment for varying stress state can only be attributed to the training of the network and the way the model has trained to generate significantly accurate results. It can be observed that the stress state of one direction does not affect the stress increment of any other direction and this is due to the construction of the neural network where every stress state is fed individually to the network.

5.2.6 Relative Contribution metric

This metric gives an estimation of the control on the inputs and outputs of the network based on the weights of the Neural network. Thus the contribution of each input can be determined by the magnitude and direction of the connecting weights. The Overall connection weights(Olden and Jackson, 2002) are used to determine the contribution of every input to the output. This neural network model consists of six parallel networks. Therefore, to obtain the overview of the relative contribution of the inputs on the output, each network is computed individually and the average contribution of all networks is obtained. Following plot represents the relative contribution of the inputs in the entire network:

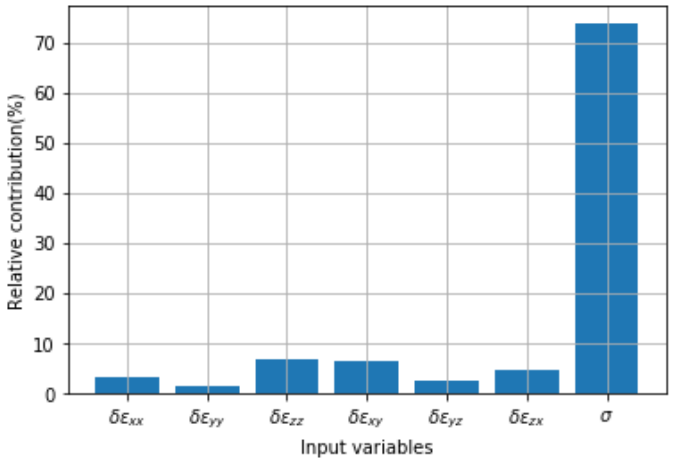


Figure 5.21: Relative contribution to the neural network

It is observed that in the devised neural network the stress state input is the major contributing factor to the output of the model with a factor of 74%. This implies that stress state as a history point has a

major influence on the neural network in order to obtain accurate results.

5.3 Summary

In this chapter, a component based approach was used to model a neural network to predict soil behavior. A soil model can be broken down into typical behaviors such as elasticity, plasticity etc. These behaviors were represented in this chapter using mechanical components like spring, dashpot and slider. This was followed by the modeling of neural network for Linear Elastic soil for which the input was strain increment and the output was stress increment. The training data was generated from CRS test, Biaxial test and DSS tests. The neural network model was then tested for accuracy and it was observed that the prediction of the model was accurate. The error in the isotropic linear elastic matrix(derived from the neural network) when compared with the theoretical matrix was insignificant. Therefore, it is concluded that the model is successful in predicting linear elastic behavior.

Following this, the Mohr Coulomb behavior was devised in a neural network using "ReLU" activation functions. The training data was generated using True Triaxial tests and DSS tests. The neural network was initially modeled in such a manner that the input to the model was strain increment and the output was stress increment. The predictions from the network diverged, implying that an additional feature was required as an input to the network. Therefore, stress state was added as an input to the network. The network was then modeled for a single direction and the response of the neurons was plotted for elastic and plastic states in order to understand inner working of the model. This was followed by including all the directions to the network and evaluating the accuracy of predictions. The neural network did produce accurate predictions, but it was observed that the network was unable to learn the behavior of zero output for a zero input. This behaviour is important because during the application of pure principal (or shear) strains, the stresses in shear (or principal) strains must be zero. Inability of the model to learn this behaviour would lead to predictions with large stresses for zero input which is physically incorrect. Therefore, a dataset for shear directions was added in each direction and to avoid over-training of the network, early stopping and dropout were added. The model was evaluated for various dropouts and 0.05 was concluded to be the best dropout for the neural network. The model was then evaluated for its generalization ability with two methods - introducing noise in the training data and varying the initial value for testing the data. It was found that the neural network has a practical applicability with noise of $\pm 6kPa$ in the training datasets (generated from laboratory experiments) and upto $3kPa$ in the testing datasets. Further, the accuracy of the model was evaluated for simpler soil behaviours such as linear elastic model. It was observed that the model was able to predict the linear elastic behaviour with a slight deviation in prediction caused due to the error accumulation from feedback loop.

A sensitivity analysis was conducted to analyze the importance of every variable in the neural network. It was observed that the relation between strain increments and stress increments mirrors the behaviour of True triaxial and DSS tests which were used to train the network. However, no physical relation was established between the stress states and the stress increments. Furthermore, through the relative contribution metric, it was observed that the stress state is the major influencing variable in the neural network which demonstrates the importance of history points in prediction of soil behaviour.

In conclusion, the neural networks were capable in predicting Linear Elastic and Elasto-plastic behaviours. Through this chapter, three research questions are answered:

1. *Could only the strain increments be used as an input to predict the stress increment or are more parameters required as the complexity of behaviour increases?*

   For Linear elastic soil model, the input of strain increment is sufficient in predicting the soil behaviour. However as the complexity of the soil model increases, for the prediction of Mohr-Coulomb model, an additional input of stress state is also required.

2. *What are the standard lab tests that must be considered such that the behaviour of every soil model is captured successfully?*

   The laboratory tests considered for Linear Elastic tests were CRS, Biaxial and DSS tests. In the case of Mohr-Coulomb model, True Triaxial test and DSS tests were used to generate training datasets for a successful neural network model. Since, in a conventional Triaxial test, the strains are applied in y-direction leading to stresses in only y-direction(in PLAXIS), it has a physical limitation in deriving a relation between strains and stresses in all directions. In case of a CRS test, it has a limitation of not reaching failure which is an important characteristic to model Mohr-Coulomb behaviour. Therefore, True Triaxial test had to be used though it is not a standard laboratory test.

3. *Could the behaviour of simpler soil models be predicted from complex soil models?*

From the analysis in this chapter, it was observed that neural network representing Mohr Coulomb behaviour was able to predict the Linear elastic soil model. Therefore, it can be concluded that simpler soil behaviours can be predicted from complex soil models. **Note**: This is true for Linear elastic and Mohr-Coulomb models. The same cannot be concluded for other models and hence will require verification.

After the successful modeling of the soil models in neural network, the next step is the implementation of neural networks in constitutive models and verification of the accuracy in a Finite Element Analysis.

Chapter 6

Finite Element Implementation of Neural networks

Finite Element Analysis is a computationally expensive method. Every iteration of the computation also required the computation of the stresses in the constitutive model through mathematical convergence methods such as Newton-Raphson method. Therefore, to minimize the amount of computation, several attempts have been made in implementing direct methods where the stresses could be predicted from the strains. At this juncture, it led to the introduction of ANN in the Finite Element model for computation. In this chapter, an attempt is made to implement the constitutive models of Linear elastic and Mohr-Coulomb behavior and then analyze the accuracy of the constitutive model by implementing in a Finite Element Analysis.

As mentioned earlier, for the calculations of the constitutive model (which would subsequently be used in finite element analysis), the computation of the material stiffness matrix is required. From Hashash et al., 2004, the material stiffness matrix can be derived from the constructed ANN model. The material stiffness matrix in the case of an isotropic linear elastic model is the elastic stiffness matrix. The idea is to implement the neural network matrix in the existing UMAT/UDSM. The calculations of the material stiffness matrix in the UMAT/UDSM are replaced with the Jacobian derived from the Neural network. In this chapter, the jacobian matrix for the modified neural network of linear elastic model is derived and implemented in ABAQUS. This is followed by the comparison of the obtained results with the theoretical results of a FE model in ABAQUS. Following this, the jacobian matrix for the Mohr-Coulomb model is derived. Subsequently, the mathematical derivation of the jacobain matrix is implemented to create a constitutive model for Mohr-Coulomb soil in PLAXIS. The results of the constitutive model are compared with an existing(theoretical) Mohr-Coulomb model in PLAXIS. Both the models are further compared by implementing them in a FE model in PLAXIS, followed by the inference made from the results.

6.1 Finite element implementation of Linear Elastic model

In the Linear Elastic model, the neural network consists of a single layer of weights with a linear activation function. Therefore the implementation of the neural network matrix in the UMAT for the Linear elastic constitutive model is considerably straight-forward where the material stiffness matrix in the UMAT for linear elastic model is replaced by the denormalized weights of neural network(derived in Chapter 5, table 5.2). The implementation of the UMAT has been done in Python where the weight matrix of the neural network has been replaced in the script for the theoretical D-matrix. Using the derived D-matrix, the UMAT for Linear elastic neural network model has been computed. For the analysis of the accuracy of the neural network model, the derived UMAT has been implemented in program 56 or $p56$ of Smith et al., 2015b. The results for the ANN implemented UMAT and the theoretical UMAT are as follows:

The integration point (nip= 1) stresses are:

Element	x-coord	y-coord	z-coord			
sig_x	sig_y	sig_z	tau_xy	tau_yz	tau_zx	
1	0.2500E+00	0.5000E+00	-0.5000E+00			
0.1117E-03	-0.1514E-02	-0.9086E-02	0.2721E-04	0.8174E-03	0.3866E-04	
2	0.2500E+00	0.5000E+00	-0.1500E+01			
0.3206E-03	-0.5392E-03	-0.6522E-02	-0.1129E-04	0.8436E-03	0.4789E-04	
3	0.2500E+00	0.1500E+01	-0.5000E+00			
-0.1047E-03	-0.1184E-02	-0.8991E-03	0.9063E-05	0.8102E-03	-0.2664E-04	
4	0.2500E+00	0.1500E+01	-0.1500E+01			
0.1726E-03	-0.8932E-03	-0.2679E-02	-0.2100E-04	0.1263E-02	0.2640E-04	
5	0.2500E+00	0.2500E+01	-0.5000E+00			
0.2972E-04	0.6960E-04	-0.8048E-04	-0.1775E-04	-0.5355E-04	0.1032E-04	
6	0.2500E+00	0.2500E+01	-0.1500E+01			
0.6262E-04	-0.4663E-03	-0.2567E-03	-0.3340E-05	0.5500E-03	-0.4843E-06	

(a) Values from Neural network implemented UMAT

The integration point (nip= 1) stresses are:

Element	x-coord	y-coord	z-coord			
sig_x	sig_y	sig_z	tau_xy	tau_yz	tau_zx	
1	0.2500E+00	0.5000E+00	-0.5000E+00			
0.1113E-03	-0.1515E-02	-0.9086E-02	0.2734E-04	0.8176E-03	0.3856E-04	
2	0.2500E+00	0.5000E+00	-0.1500E+01			
0.3205E-03	-0.5395E-03	-0.6521E-02	-0.1129E-04	0.8439E-03	0.4784E-04	
3	0.2500E+00	0.1500E+01	-0.5000E+00			
-0.1046E-03	-0.1185E-02	-0.8993E-03	0.8984E-05	0.8106E-03	-0.2672E-04	
4	0.2500E+00	0.1500E+01	-0.1500E+01			
0.1726E-03	-0.8936E-03	-0.2680E-02	-0.2097E-04	0.1263E-02	0.2637E-04	
5	0.2500E+00	0.2500E+01	-0.5000E+00			
0.2987E-04	0.6990E-04	-0.8058E-04	-0.1787E-04	-0.5362E-04	0.1042E-04	
6	0.2500E+00	0.2500E+01	-0.1500E+01			
0.6259E-04	-0.4663E-03	-0.2570E-03	-0.3323E-05	0.5502E-03	-0.4988E-06	

(b) Values from theoretical/existing UMAT

sig_x	sig_y	sig_z	tau_xy	tau_yz	tau_zx
4.00E-07	1.00E-06	0.00E+00	-1.30E-07	-2.00E-07	9.00E-08
1.00E-07	3.00E-07	-1.00E-06	0.00E+00	-3.00E-07	4.00E-08
-2.00E-07	1.00E-06	2.00E-07	9.40E-08	-4.00E-07	7.00E-08
0.00E+00	3.00E-07	1.00E-06	-2.00E-08	0.00E+00	2.00E-08
-6.00E-08	-2.40E-07	1.70E-07	1.40E-07	7.00E-08	-9.00E-08
5.00E-08	1.00E-07	3.00E-07	-1.10E-08	-2.00E-07	2.07E-08

(c) Error difference in values

Figure 6.1: Comparison of ANN UMAT and theoretical UMAT

It can be observed that the error of the model is in the order of 10^{-7} which is a significantly small error. Based on this minuscule error, the model can be considered to be accurate. Hence, the implementation of the constitutive model devised for Linear Elastic soil, in Finite Element Model was successful.

6.2 Implementation of Mohr Coulomb model

After the implementation of the Linear elastic model in a finite element analysis, the next step involves the finite element implementation of the Mohr-Coulomb model. Theoretically, for a constitutive model of the Mohr-Coulomb soil, two parts of the material stiffness matrix - the elastic and the plastic part are considered. Based on the plastic yield criterion, the elastic and the plastic part of the matrix are taken into account, which have undergone the Newton Raphson method for convergence. In the case of the ANN, the matrix would be directly taken using the Jacobian matrix. The Jacobian matrix or the material stiffness matrix for the Mohr Coulomb model is derived in the following section.

6.2.1 Stiffness matrix computation

Consider for a given node of a hidden layer H_i, and given input node X_i, weight W_{ij}, bias b_i and σ representing the activation function, the output from the hidden layer can be expressed as:

$$H_i = \sigma(W_{ij}X_i + b_i) \tag{6.1}$$

For the Mohr Coulomb model, with the input strain increment vector $\Delta\epsilon_i$ and stress state σ_i. The inputs are scaled as follows:

$$\Delta\epsilon_i^{NN} = \frac{\Delta\epsilon_i - S_{min}^{\Delta\epsilon}}{S_{max}^{\Delta\epsilon} - S_{min}^{\Delta\epsilon}} \tag{6.2}$$

$$SV_i^{NN} = \frac{SV_i - S_{min}^{SV}}{S_max^{SV} - S_{min}^{SV}} \tag{6.3}$$

where S_{min}^{ϵ} and S_{min}^{SV} represent the minimum values of the strain increment vector and stress state vector respectively. Similarly, S_{max}^{ϵ} and S_{max}^{SV} represent the maximum values of the strain increment vector and stress state vector respectively. For the hidden layer, B_i, the equation is as follows:

$$B_i = ReLU\left[\beta\left[\sum_{j=1}^{N\Delta\epsilon} W_{ij}^{B\Delta\epsilon}\Delta\epsilon_j^{NN} + b_j^{B\Delta\epsilon} + \sum_{j=1}^{NSV} W_{ij}^{BSV}SV_j^{NN} + b_j^{BSV}\right]\right] \tag{6.4}$$

where

$W_{ij}^{B\Delta\epsilon}$ = connection weight between the input node $\delta\epsilon_j$ and the hidden layer node B_i
Similarly W_{ij}^{BSV} = connection weight between the input node SV_j and the hidden layer node B_i
β = constant for the Linear activation function

For the second ReLU layer,

$$C_i = ReLU\left[\beta\left[\sum_{j=1}^{N\Delta\epsilon} W_{ij}^{CB}B_j^{NN} + b_j^{CB}\right]\right] \tag{6.5}$$

For the output stress vector, $\delta\sigma_i^{NN}$,

$$\Delta\sigma_i^{NN} = Linear\left[\beta\left[\sum_{j=1}^{NC} W_{ij}^{C\Delta\sigma}C_j\right]\right] \tag{6.6}$$

De-normalising, the stress vector implies:

$$\Delta\sigma_i = (S_{max}^{\Delta\sigma} - S_{min}^{\Delta\sigma})\Delta\sigma_i^{NN} + S_{min}^{\Delta\sigma} \tag{6.7}$$

where $S_{max}^{\Delta\sigma}$ and $S_{min}^{\Delta\sigma}$ represent the maximum and minimum values for the stress increment vector. The jacobian is represented as:

$$^{n+1}J = \frac{\partial\left(^{n+1}\Delta\sigma\right)}{\partial\left(^{n+1}\Delta\epsilon\right)} \tag{6.8}$$

From equation (6.8),

$$\begin{aligned}\frac{\partial^{n+1}\Delta\sigma_i}{\partial^{n+1}\Delta\epsilon_j} &= \frac{\partial^{n+1}\Delta\sigma_i}{\partial^{n+1}\Delta\sigma_i^{NN}}\frac{\partial^{n+1}\Delta\sigma_i^{NN}}{\partial^{n+1}\Delta\epsilon_j} \\ &= \frac{\partial^{n+1}\Delta\sigma_i}{\partial^{n+1}\Delta\sigma_i^{NN}}\frac{\partial^{n+1}\Delta\sigma_i^{NN}}{\partial^{n+1}\Delta\epsilon_j^{NN}}\frac{\partial^{n+1}\Delta\epsilon_j^{NN}}{\partial^{n+1}\Delta\epsilon_j}\end{aligned} \tag{6.9}$$

However,

$$\begin{aligned}S_{max}^{\Delta\sigma} - S_{min}^{\Delta\sigma} &= \frac{\partial^{n+1}\Delta\sigma_i}{\partial^{n+1}\Delta\sigma_i^{NN}} \\ \frac{1}{S_{max}^{\Delta\epsilon} - S_{min}^{\Delta\epsilon}} &= \frac{\partial^{n+1}\Delta\epsilon_j^{NN}}{\partial^{n+1}\Delta\epsilon_j}\end{aligned} \tag{6.10}$$

Hence, from equation (6.9),

$$\frac{\partial^{n+1}\Delta\sigma_i}{\partial^{n+1}\Delta\epsilon_j} = \frac{S_{max}^{\Delta\sigma} - S_{min}}{S_{max}^{\Delta\epsilon} - S_{min}^{\Delta\epsilon}}\frac{\partial^{n+1}\Delta\sigma_i^{NN}}{\partial^{n+1}\Delta\epsilon_j^{NN}} \tag{6.11}$$

From equation (6.11),

$$\frac{\partial^{n+1}\Delta\sigma_i^{NN}}{\partial^{n+1}\Delta\epsilon_j^{NN}} = \sum_{l=1}^{NC}\left(\frac{\partial^{n+1}\Delta\sigma_i^{NN}}{\partial^{n+1}C_l}\sum_{k=1}^{NB}\left(\frac{\partial^{n+1}C_l}{\partial^{n+1}B_k}\frac{\partial^{n+1}B_k}{\partial^{n+1}\Delta\epsilon_j^{NN}}\right)\right) \tag{6.12}$$

For a ReLU activation function, the derivative implies:

$$\frac{\partial ReLU(f(x))}{\partial x} = step(f|x|)\frac{\partial f(x)}{\partial x} \tag{6.13}$$

Hence,

$$\frac{\partial^{n+1} C_l}{\partial^{n+1} B_k} = step(C_l)\beta w_{lk}^{CB} \tag{6.14}$$

$$\frac{\partial^{n+1} B_k}{\partial^{n+1} \Delta\epsilon_j^{NN}} = step(B_k)\beta w_{kj}^{B\Delta\epsilon} \tag{6.15}$$

Similarly,

$$\begin{aligned}\frac{\partial^{n+1}\Delta\sigma_i^{NN}}{\partial^{n+1} C_l} &= \frac{\partial}{\partial^{n+1} C_l}\left(\beta\left[\sum_{l=1}^{NC} w_{il}^{\Delta\sigma C}\left({}^{n+1}C_l\right)\right]\right) \\ &= \beta w_{ik}^{\Delta\sigma C}\end{aligned} \tag{6.16}$$

Hence, the Jacobian for the neural network can be expressed as:

$$\frac{\partial^{n+1}\Delta\sigma_i}{\partial^{n+1}\Delta\epsilon_j} = \frac{S_{max}^{\Delta\sigma} - S_{min}}{S_{max}^{\Delta\epsilon} - S_{min}^{\Delta\epsilon}}\beta^4 \sum_{l=1}^{NC}\left(w_{il}^{\Delta\sigma C}\sum_{k=1}^{NB}\left(step(C_l)w_{lk}^{CB}step(B_k)w_{kj}^{B\Delta\epsilon}\right)\right) \tag{6.17}$$

Considering, the value of $\beta = 1$, this implies

$$\frac{\partial^{n+1}\Delta\sigma_i}{\partial^{n+1}\Delta\epsilon_j} = \frac{S_{max}^{\Delta\sigma} - S_{min}}{S_{max}^{\Delta\epsilon} - S_{min}^{\Delta\epsilon}} \sum_{l=1}^{NC}\left(w_{il}^{\Delta\sigma C}\sum_{k=1}^{NB}\left(step(C_l)w_{lk}^{CB}step(B_k)w_{kj}^{B\Delta\epsilon}\right)\right) \tag{6.18}$$

The Jacobian matrix thus computed, has been used in the UDSM of Mohr Coulomb soil for stress computation. The accuracy of the derived UDSM has been verified in the Soil test tab of PLAXIS. A similar kind of a data(as for the neural network) with 100 phases and a random strain increment from a normal distribution of $-5e-4$ to $2e-4$ for true triaxial test and direct shear test have been generated in the form of a .vls file. The accuracy of the neural network model has then been verified with respect to the theoretical Mohr Coulomb model through the generated files in the "General" tab of Soil test in PLAXIS 3D. For the computation of stresses from the strain increments the Jacobian matrix computed above has been used through the following equations:

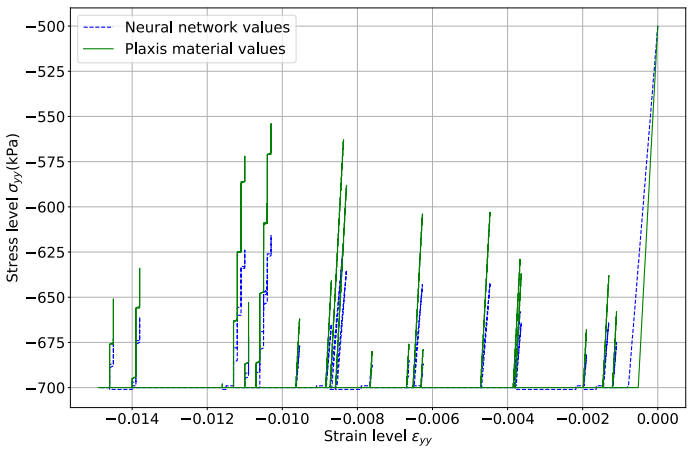
$$\sigma^t = \sigma_0^{t-1} + D * d\epsilon^t \tag{6.19}$$

and

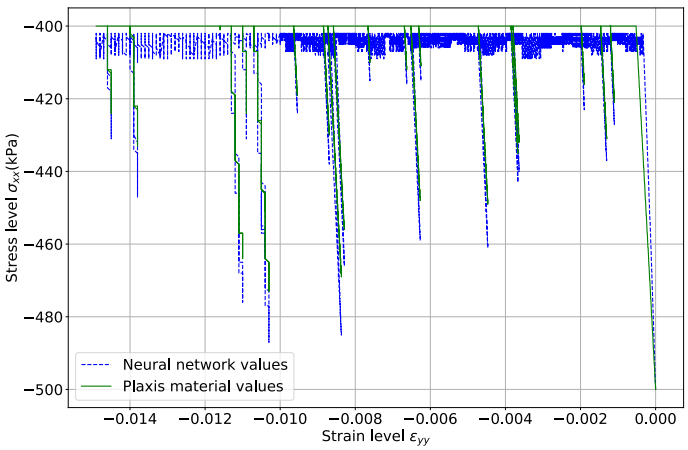
$$\sigma_0^t = \sigma^t \tag{6.20}$$

where σ^t is the stress state of the current step, σ_0^{t-1} represents the previous stress state, D represents the Jacobian or the material stiffness matrix and $d\epsilon^t$ is the strain increment of the current step. After computation of stress state in the current stress state, the previous stress state is updated which is represented in equation (6.20).

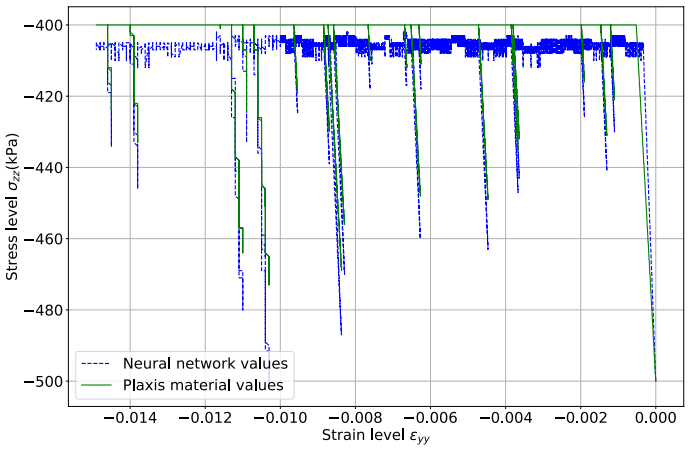
Following are the results from true triaxial test and the DSS test for the neural network model plotted along with the theoretical Mohr-Coulomb model to visualize the accuracy of the Neural network model:



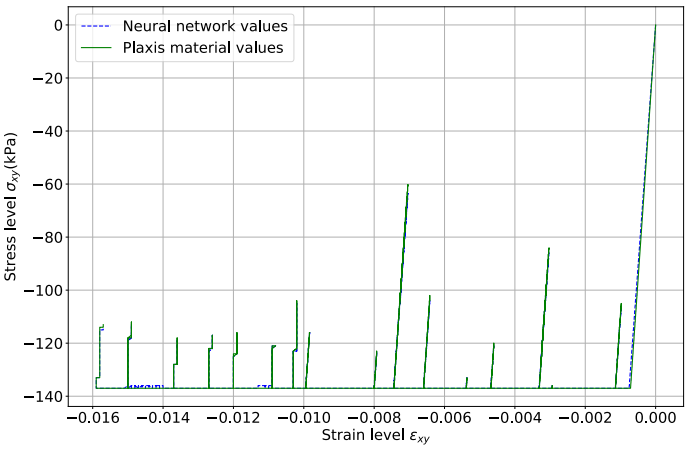
(a) Strain vs Stress in yy-direction



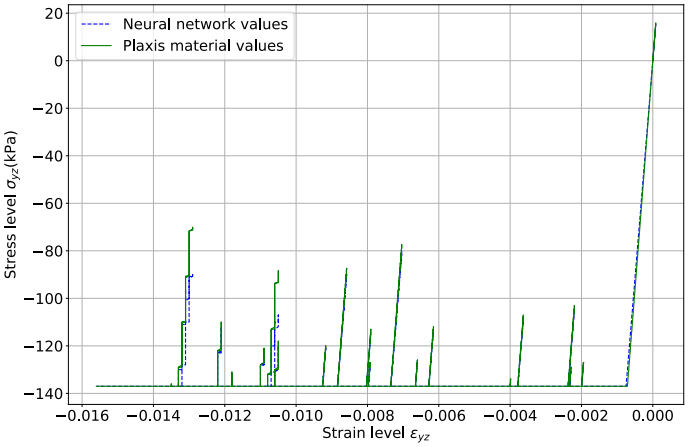
(b) Strain vs Stress in xx-direction



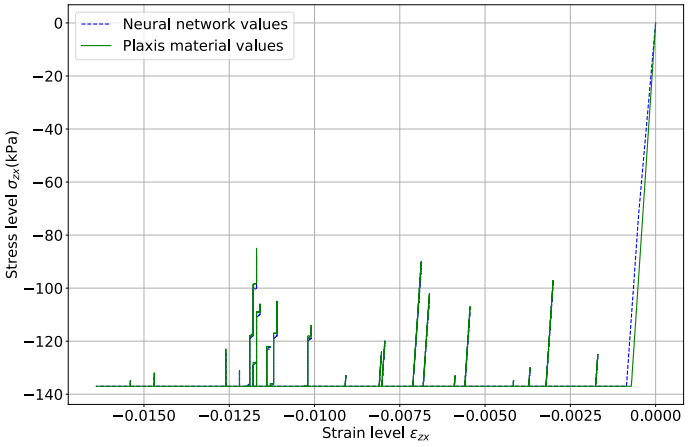
(c) Strain vs Stress in zz-direction



(d) Strain vs Stress in xy-direction



(e) Strain vs Stress in yz-direction



(f) Strain vs Stress in zx-direction

Figure 6.2: Comparison of Neural network constitutive model with Mohr-Coulomb model in PLAXIS

It can be observed that the Neural network is able to predict the elasto-plastic behaviour of the soil, but the prediction of the model at the point of plasticity and during re-loading is not accurate. This is due to the errors in the prediction of the neural network which has led to errors in the Jacobian matrix and eventually the computed stresses. The errors in the Neural network constitutive model with respect to the Mohr-Coulomb model are as follows:

Stress increment	Mean Absolute error(kPa)
σ_{xx}	4.505
σ_{yy}	8.565
σ_{zz}	6.524
σ_{xy}	0.0409
σ_{yz}	0.67
σ_{zx}	0.14

Table 6.1: Absolute error for Neural network constitutive model with respect to the Mohr-Coulomb model

Despite these errors, it can be concluded that the constitutive model derived from neural networks is able to predict the elasto-plastic behavior because, the model is capable of understanding the trend of elasto-plastic soil in unloading-reloading conditions, as seen from the plots above. Further an attempt is made to implement the constitutive model in a finite element model.

6.2.2 Implementation in FE model

Implementation of ANN constitutive model has been carried out by constructing a finite element model in PLAXIS. With regard to the finite element model, the idea has been to initially consider a simple model and subsequently, based on the accuracy of the model, implement the constitutive model in a complex FEM scenario such as settlement in building foundation. For the initial case, a block of soil has been considered with the dimensions of $12m * 8m * 12m$ in a dry condition. The model was then fixed in the z-direction in the bottom and subjected to a surface load of $-1kN/m^2$ on the top surface. Following is the representation of the model:

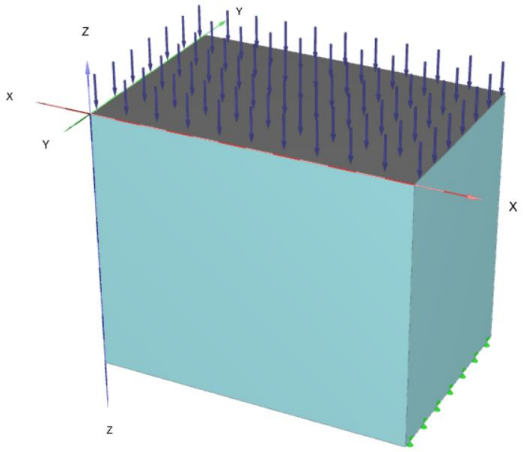


Figure 6.3: Finite Element Model for implementation of Neural network constitutive model

Two phases have been created - with an initial phase, only the soil is activated. In the loading phase the displacement condition and the loads were activated. The expected result is the compression of the soil under load, leading to a deformation. However, it has been observed from the results that, during calculation of the loading phase, a convergence error occurs in the model. To further investigate the issue, the material stiffness matrix has been printed as an output and it was observed that certain columns in the matrix are computed to be zero.

```
D =
 0.5212E+05 -0.9555E+05 -0.1784E+05  0.0000E+00  0.0000E+00  0.0000E+00
 0.3663E+05  0.1269E+06 -0.1066E+05  0.0000E+00  0.0000E+00  0.0000E+00
 0.2409E+05 -0.1568E+06  0.2406E+05  0.0000E+00  0.0000E+00  0.0000E+00
-0.8387E+05 -0.1720E+05 -0.3691E+04  0.0000E+00  0.0000E+00  0.0000E+00
 0.3688E+05  0.2847E+05 -0.6083E+05  0.0000E+00  0.0000E+00  0.0000E+00
 0.1904E+05 -0.2632E+04  0.8371E+05  0.0000E+00  0.0000E+00  0.0000E+00
```

Figure 6.4: Obtained D-matrix from finite element implementation

In PLAXIS, implementing the neural network constitutive model led to the assignment of strain increment values and eventually the stress state to zero. Therefore during the computation of the stiffness matrix, this has led to singularity of the matrix i.e., the determinant of the matrix equals zero. The reason for the assignment of zeroes to the strain increments was however unknown. The material stiffness matrix is used in the computation of the global error which is represented by the following equation:

$$Globalerror = \frac{\sum ||Outofbalancenodalforces||}{\sum ||Activeforces|| + CSP.||Inactiveloads||} \tag{6.21}$$

where *Out of balance nodal forces* represent the difference between the external loads and the forces in equilibrium with the current stresses, *Active forces* represent the current calculation and previous calculation phase and *Inactive loads* are the loads of the previous calculation phase. CSP which is the current

value of the stiffness parameter representing the measure for amount of plasticity occurring during the calculation. It is represented by following equation(Brinkgreve et al., 2013):

$$CSP = \int \frac{\Delta\epsilon.\Delta\sigma}{\Delta\epsilon D^e \Delta\epsilon} \tag{6.22}$$

The model consists of corners where the element size could be zero, consequently leading to a convergence error. In order to counter this, a refined mesh has been examined. However, the error of convergence could not be eliminated. It has been observed that for a very minute surface load of $1 * 10^{-9} kN/m^2$, the model was able to converge successfully, resulting in a total displacement of $0.1843 * 10^{-15} m$. The same amount of load of $1 * 10^{-9} kN/m^2$ was applied on the Finite Element model (using the theoretical Mohr-Coulomb constitutive model). The displacement was observed to be $0.01486 * 10^{-12} m$. For the Mohr-Coulomb soil model, the implementation of the neural network in a constitutive model has been successful. However, implementation of the same in a Finite element model would require a different approach that may need to be devised in order to converge to an apt solution.

6.3 Summary

The chapter begins with the implementation of neural networks in Linear elastic constitutive model. The computed material stiffness matrix in the previous chapter, has been replaced with the theoretical D-matrix of Linear Elastic UMAT subroutine. Subsequently, to analyse the accuracy of the linear elastic model, it has been implemented in a finite element model, $p56$ of Smith et al., 2015b. It has been observed that the maximum error in the model is $1 * 10^{-6}$ which is an insignificant error. Therefore, it has been concluded that the implementation of the Neural network in Finite element model for Linear elastic soil has been successful.

This was followed by the implementation of neural network in a constitutive model for Mohr Coulomb soil. The Jacobian for the computation of the material stiffness matrix has been derived by computing the partial derivatives of the neural network layers. The derived Jacobian has been subsequently used to formulate the UDSM(for PLAXIS) which represents the constitutive model for Mohr Coulomb soil. To analyze the accuracy of the constitutive model, it has been implemented as a soil model in PLAXIS Soil test for True Triaxial Test and DSS tests. The obtained stresses were compared with the stresses for theoretical Mohr-Coulomb soil model(tested in PLAXIS Soil test). It has been observed that the maximum absolute error was $8.565 kPa$ which is a significant error. The error could however be minimized by improving the performance of the Neural network. From the plots, it has been inferred that the model was able to predict the Mohr Coulomb behaviour with a decent accuracy.

The constitutive model was then implemented in Finite Element analysis, initially for a simple FE model in PLAXIS. Two phases of an initial phase and a loading phase were considered. During the loading phase, the model led to a convergence error. Upon investigation, it was observed that the material stiffness matrix led to a singular matrix(zero determinant). The material matrix when used for the computation of global error, therefore, led to an infinite error which was the cause for the convergence error. The mesh for the FE model was subsequently refined to eliminate any sharp corners where the meshes intersect, leading to a singular matrix. However, the problem of convergence error still remained. It was observed that for a minute load of $1 * 10^{-9} kN/m^2$, the model was able to converge.

Answering the final research question of the thesis: *Could soil models like Elasto-plastic model which require computation of linear elastic matrix and plastic flow criterion be modeled using neural networks?*
It is possible to devise a constitutive model using neural networks for soil models like Elasto-plastic model. Caution needs to be taken in terms of prediction of the neural network, making sure that the model is able to learn every characteristic of the soil behavior with a significant accuracy; failing which, the constitutive model would lead to undesired results. It is therefore concluded that though the neural network implemented constitutive model was a success, more work is required in the implementation of the constitutive model in Finite Element Analysis.

Chapter 7

Conclusions

The objective of the thesis was to determine if every typical soil behaviour could be predicted through a neural network model and further be implemented in a constitutive model. The primary soil behaviours discussed in this thesis are elasticity and plasticity. The path towards the objective began with the idea of developing a generic neural network model which could predict the natural phenomenon of soil when subjected to a load. This was further elaborated with the implementation of neural networks to predict a typical soil behaviour such as elasticity and plasticity by considering a component based approach. This was further developed to implement the neural network in Finite Element model.

The approach to the generic neural network was developed to predict the behaviour of Linear elastic soil. Two types of tests- standard and non-standard lab tests were chosen for the purpose. The accuracy of prediction of both, the standard lab data and the non-standard lab data from the models trained with similar kind of data was high. However, prediction of one type of model with respect to the other led to inaccuracies in the models due to the fact that, the stress paths created in the two types of tests were completely different from each other. The neural network trained with the standard data was implemented in Finite element analysis by developing a material stiffness matrix (for a constitutive model) from a Jacobian computed from the neural network. It was observed that the stiffness matrix did not coincide with the isotropic linear elastic matrix. Since, the weights of the neural networks were not appropriately constrained, though the prediction was accurate, the solution of the neural network(material stiffness matrix) had infinite possibilities for a converging solution. It was concluded that the approach to create a generic neural network cannot be used due to the following limitations: (1) Standard and non-standard tests could not be implemented in a practical laboratory scenario. (2) The method in which non-standard and standard lab tests were implemented, simulation of anisotropic and isotropic behaviour of soil respectively, was not guaranteed by these methods. This implied that prediction of anisotropic behaviour through neural networks is not possible at the current stage and further research would be required to model the same.

An alternative approach to the implementation of neural network was therefore made. Standard laboratory tests such as CRS test, Biaxial test and DSS tests were necessitated to encompass every soil behavior. In order that the neural network could predict typical behaviors such as elasticity and plasticity, a component based approach was implemented. For the linear elastic neural network, strain increments and stress increments respectively were taken as inputs and outputs. The weights for the network were constrained such that the stress-strain relations in an isotropic linear elastic matrix were kept intact. The neural network was subsequently extended to predict the plastic(Mohr-Coulomb) behaviour of soil. In the case of Mohr-Coulomb model, for creating datasets using a conventional Triaxial test, the strains are applied in y-direction leading to stresses in only y-direction(in PLAXIS), leading to a physical limitation in deriving a relation between strains and stresses in all directions. In order toconsider a CRS test, it has a limitation of not reaching failure which is an important characteristic to model Mohr-Coulomb behaviour. Therefore, True Triaxial test had to be adopted, though it was known that the test was not fully developed.

Initially, the Mohr Coulomb neural network was developed with an input of strain increment to predict stress increment. When the elastic and plastic behaviors in Mohr-Coulomb model were to be studied, single input of strain increment was seen to be insufficient to converge to an accurate prediction. As the complexity of the soil models increased, the inputs also had to be gradually increased, in order to capture the complete soil behavior. Two inputs, viz. strain increment and stress state were required to converge to an accurate prediction model for Mohr-Coulomb soil. However, the neural network was unable to learn the characteristic of zero outputs for zero inputs. Inability to learn this behaviour would lead the model to

produce stresses in directions where there is no strain applied, which is phsyically implausible. Therefore, an extra dataset in each shear direction was added. In order to avoid overtraining of the network and improve the generalization ability, a dropout layer and early stopping were introduced. A search algorithm was created to evaluate the best dropout value for the network. With a dropout of 0.05, the model was able to learn every characteristic with significant accuracy. For practical applicability of the neural network model, generalization ability was subsequently analyzed in two ways: (1) Introducing a noise in the training datasets (2) Varying the initial stresses in the testing data. It was observed that the neural network produced accurate results until a noise of $\pm 6kPa$ in training datasets and a deviation in initial stresses upto $3kPa$. Further, a sensitivity analysis and relative contribution metric were carried out in order to understand the dependence of network on input variables and to find the governing input variable in the model. Through the sensitivity analysis, it was observed that the strain increments and the stress increments replicated the behaviour of datasets(true triaxial test and DSS tests) with which the neural network was trained. Though strain increment is a necessary input, through the relative contribution metric, it was assessed that the stress state input was a major contributing factor to the prediction model.

The successful prediction of the Mohr-Coulomb soil behavior was followed-up by improving the robustness of the neural network in order that the typical behavior of plasticity was well-learnt by the model. Considering the complexity of Mohr-Coulomb model as compared to Linear elastic soil model, the accuracy of prediction of linear elastic behaviour in Mohr-Coulomb neural network model was examined. It was observed that the neural network was able to predict the behavior with a minor error due to the error accumulation of the feedback loop during testing. Therefore, future work for further models might mandate more study into the cause, since the increasing complexity of the model might not guarantee the prediction of simpler models.

With the successful implementation of the neural network, the next main objective was its implementation into the Finite Element model. By using mathematics from the literature, the material stiffness matrix was derived from the neural network model. Implementation of the Neural network in a constitutive model for Linear elastic soil was analyzed by comparing the linear elastic matrices(obtained from network and theoretical matrix). Further, the constitutive model was subjected to a Finite Element analysis, and the model was found to be accurate. In Mohr-Coulomb model, the material stiffness matrix was derived for the constitutive model and implemented in a UDSM(used in PLAXIS). The constitutive model was then analyzed for accuracy in PLAXIS soil test. Although it was observed that the model had certain errors, the model was able to predict the Mohr Coulomb behavior with a decent accuracy. Implementation of the model further for multiple elements however, led to convergence error in the Finite Element model, due to occurrence of singularity in the matrix which lead to divergence of global error in the model. One possible reason for such singularity of the matrix could be the mesh convergence at the corners. A fine mesh for the model could possibly help in reducing corners in the Finite element model, thus avoiding the singularity of the material stiffness matrix, and preventing any convergence error.

The main research question of the thesis, thus can conclusively be answered that, it is possible to model the typical behaviour of soil through neural networks and implement the same in a constitutive model. Further work may however be required for implementation of the constitutive model in the Finite Element Engine.

Chapter 8

Lessons Learnt and Future Recommendations

Lessons learnt from the project which are considered useful for subsequent projects are stated as under:

- In this thesis, a combination of laboratory tests was made use of, in order to encompass every behaviour of Mohr-Coulomb model. However due to the physical limitations of the tests, True Triaxial Test had to be taken into account, which itself was not a fully developed test. Therefore, though accurate results were obtained from the model trained from these, a detailed investigation may have to be carried out to find a standard test for usage.
- For the neural network model of Mohr Coulomb soil, the stress states were a single input to the entire network(stress increment in one direction was dependent on the stress state of all directions). This resulted in inaccuracy of the model. Therefore, the neural network architecture necessitated required the stress states to be considered separately for every direction, to converge to a significantly accurate solution.
- For the constitutive model using the Jacobian matrix, the solution would converge to accurate stresses only and only if, the secondary values in the prediction i.e., for example, the shear stresses in the triaxial test or principal stresses in the shear tests converge to zero.

8.1 Future Recommendations

Although the thesis was able to answer the research questions posed in the introduction, there are recommendations with respect to the Neural Networks and constitutive modelling which could be considered:

- For the implementation of the constitutive model, an attempt could be made to implement the model for a simple structure with consisting of 2 elements to resolve the issue of convergence of the matrix.
- The stresses in the constitutive model could be computed using weights and biases of the network, further computing the secant matrix for the material stiffness matrix.
- The neural network could be expanded to accommodate for other typical soil behaviours such as creep, hardening etc with the inclusion of more components into the network inspired from Maxwell models using activation functions.

Bibliography

Abelev, A., & Lade, P. (2004). Characterization of failure in cross-anisotropic soils. *Journal of Engineering Mechanics*, *130(5)*, 599–606.

Alhussaini, M. (1971). Investigation of plane strain shear testing. *U.S.Army Engineer Waterways Experiment Station, Vicksburg, Mississippi*, *Report I*.

Allamy, H., & Khan, R. Z. (2014). Methods to avoid over-fitting and under-fitting in supervised machine learning (comparative study), 163–172. https://doi.org/10.3850/978-981-09-5247-1_017

Berhane, G. (2006). Constitutive soil models and soil parameters. *Excavations and foundations in soft soils* (pp. 57–116). Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-32895-5_3

Brinkgreve, R., Engin, E., & Swolfs, W. (2013). Plaxis 3d 2013 user manual. *Plaxis bv, Delft*.

Britto, A., Savvidou, C., Gunn, M., & Booker, J. (1992). Finite element analysis of the coupled heat flow and consolidation around hot buried objects. *Soils and Foundations*, *32*(1), 13–25. https://doi.org/https://doi.org/10.3208/sandf1972.32.13

Bruno, D., Barca, E., Goncalves, R., Queiroz, H., Berardi, L., & Passarella, G. (2018). Linear and evolutionary polynomial regression models to forecast coastal dynamics: Comparison and reliability assessment. *Geomorphology*, *300*, 128–140. https://doi.org/10.1016/j.geomorph.2017.10.012

Corfdir, A., & Sulem, J. (2008). Comparison of extension and compression triaxial tests for dense sand and sandstone. *Acta Geotechnica*, *3*, 241–246. https://doi.org/10.1007/s11440-008-0068-x

Coulomb, C. (1973). Essai sur une application des regles de maximis et minimis a quelques problemes de statique relatifs a l'architecture (essay on maximums and minimums of rules to some static problems relating to architecture).

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS)*, *2*(4), 303–314. https://doi.org/10.1007/BF02551274

DataFlair. (2019). *Artificial neural networks for machine learning – every aspect you need to know about*. https://data-flair.training/blogs/artificial-neural-networks-for-machine-learning/ (accessed: 23.05.2021)

Drakos, S., & Pande, G. (2015). On neural network constitutive models for geomaterials. *Journal of Civil Engineering Research*, *5(5)*, 106–113. https://doi.org/10.5923/j.jce.20150505.02

Drakos, S. (2008). Applications of artificial intelligence in constitutive modelling of soils. *Swansea University*, *Thesis*. http://cronfa.swan.ac.uk/Record/cronfa42782

Duong, A. (2019). *Keras callbacks explained in three minutes*. https://www.kdnuggets.com/2019/08/keras-callbacks-explained-three-minutes.html (accessed: 11.06.2021)

Erhunmwun, I., & Ikponmwosa, U. (2017). Review on finite element method. *Journal of Applied Sciences and Environmental Management*, *21(5)*, 999–1002. https://doi.org/10.4314/jasem.v21i5.30

Feng, X., & Yang, C. (2001). Genetic evolution of nonlinear material constitutive models. *Computer Methods in Applied Mechanics and Engineering*, *190(45)*, 5957–5973. https://doi.org/10.1016/S0045-7825(01)00207-9

Gao, Z., & Zhao, J. (2013). Evaluation on failure of fiber-reinforced sand. *Journal of Geotechnical and Geoenvironmental Engineering*, *139(1)*. https://doi.org/10.1061/(ASCE)GT.1943-5606.0000737

Georgevici, A., & Terblanche, M. (2019). Neural networks and deep learning: A brief introduction. *Intensive Care Medicine*, *45*, 712–714. https://doi.org/10.1007/s00134-019-05537-w

Ghaboussi, J., Garrett Jr., J., & Wu, X. (1991). Knowledge-based modeling of material behavior with neural networks. *Journal of Engineering Mechanics, 117(1)*. https://doi.org/10.1061/(ASCE)0733-9399(1991)117:1(132)

Ghaboussi, J., & Sidarta, D. (1998a). Constitutive modeling of geomaterials from non-uniform material tests. *Computers and Geotechnics, 22(1)*, 53–71. https://doi.org/10.1016/S0266-352X(97)00035-9

Ghaboussi, J., & Sidarta, D. (1998b). New nested adaptive neural networks (nann) for constitutive modeling. *Computers and Geotechnics, 22(1)*, 29–52. https://doi.org/10.1016/S0266-352X(97)00034-7

Gonzalez, J. (2000). Experimental and theoretical investigation of constant rate of strain consolidation. *Department of Civil and Environmental Engineering, Massachusetts Institute Of Technology, M.Sc Thesis*.

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

Gulikers, T. (2018). An integrated machine learning and finite element analysis framework, applied to composite substructures including damage. *Faculty of Aerospace Engineering, Delft University of Technology, M.Sc Thesis*.

Hashash, Y. M. A., Jung, S., & Ghaboussi, J. (2004). Numerical implementation of a neural network based material model in finite element analysis. *International Journal for Numerical Methods in Engineering, 59*, 989–1005. https://doi.org/10.1002/nme.905

Hibi, Y. (2008). Formulation of a dusty gas model for multi-component diffusion in the gas phase of soil. *Soils and Foundations, 48*(3), 419–432. https://doi.org/https://doi.org/10.3208/sandf.48.419

Hwu, C. (2010). Linear anisotropic elastic materials. *Anisotropic elastic plates* (pp. 1–27). Springer US. https://doi.org/10.1007/978-1-4419-5915-7_1

Javadi, A. A., Tan, P., & Zhang, M. (2003). Neural network for constitutive modelling in finite element analysis. *University of Exeter*, Report. https://olemiss.edu/sciencenet/trefftz/Trefftz/Exeter/Javadi.pdf

Javadi, A., Ahangar-Asr, A., & Faramarzi, A. (2009). An artificial intelligence based finite element method. *ISAST Transactions on Computers and Intelligent Systems, 1*, 1–7. https://www.researchgate.net/publication/260125327_An_Artificial_Intelligence_Based_Finite_Element_Method

Javadi, A., & Rezania, M. (2008). A new approach to constitutive modelling of soils in finite element analysis using evolutionary computation. *Intelligent Computing in Engineering*. https://www.researchgate.net/publication/319015819_A_New_Approach_to_Constitutive_Modelling_of_Soils_in_Finite_Element_Analysis_using_Evolutionary_Computation

Jung, S., & Ghaboussi, J. (2006). Neural network constitutive model for rate-dependent materials. *Computers and Geotechnics, 84(15-16)*, 955–963. https://doi.org/10.1016/j.compstruc.2006.02.015

Kaselow, A. (2004). *The stress sensitivity approach: Theory and application* (Doctoral dissertation). http://dx.doi.org/10.17169/refubium-14552

Khan, R. (2019). *Nothing but numpy: Understanding and creating neural networks with computational graphs from scratch*. https://www.kdnuggets.com/2019/08/numpy-neural-networks-computational-graphs.html (accessed: 30.05.2021)

Ko, H.-Y., & Scott, R. F. (1967). A new soil testing apparatus. *Géotechnique, 17*(1), 40–57. https://doi.org/10.1680/geot.1967.17.1.40

Krose, B., & Smagt, P. (1993). An introduction to neural networks. *Journal of Computer Science*. https://www.researchgate.net/publication/272832321_An_introduction_to_neural_networks

Labuz, J., & Zang, A. (2012). Mohr–coulomb failure criterion. *Rock Mechanics and Rock Engineering volume, 45*, 975–979. https://doi.org/10.1007/s00603-012-0281-7

Lade, P., & Duncan, J. (1973). Cubical triaxial tests on cohesionless soil. *Journal of the Soil Mechanics and Foundations Division, 99(10)*, 793–812.

Mandal, B., & Chakrabarti, A. (2017). Simulating progressive damage of notched composite laminates with various lamination schemes. *International Journal of Applied Mechanics and Engineering, 22*, 333–347. https://doi.org/10.1515/ijame-2017-0020

Murakami, A., Wakai, A., & Fujisawa, K. (2010). Numerical methods. *Soils and Foundations, 50*(6), 877–892. https://doi.org/10.3208/sandf.50.877

Nguyen, N., & Waas, A. (2016). Nonlinear, finite deformation, finite element analysis. *Zeitschrift für angewandte Mathematik und Physik volume, 67*. https://doi.org/10.1007/s00033-016-0623-5

Nielsen, M. (2019). Neural networks and deep learning. http://neuralnetworksanddeeplearning.com/

Olden, J., & Jackson, D. (2002). Illuminating the "black box": A randomization approach for understanding variable contributions in artificial neural networks. *Ecological Modelling, 154(1)*, 135–150. https://doi.org/10.1016/S0304-3800(02)00064-9

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research, 12*, 2825–2830.

Penumadu, R., D.and Zhao. (1999). Triaxial compression behavior of sand and gravel using artifcial neural networks (ann). *Computers and Geotechnics, 24*, 207–230. https://doi.org/10.1016/S0266-352X(99)00002-6

Popa, H., & Batali, L. (2010). Using finite element method in geotechnical design. comparison between soil constitutive laws and case study. *Proceedings of the 3rd WSEAS International Conference on Finite Differences - Finite Elements - Finite Volumes - Boundary Elements*, 228–233. https://doi.org/10.5555/1844416.1844458

Poran, C. J., & Rodriguez, J. A. (1992). Finite element analysis of impact behavior of sand. *Soils and Foundations, 32*(4), 68–80. https://doi.org/https://doi.org/10.3208/sandf1972.32.4_68

Rees, S. (2013). *Part one: Introduction to triaxial testing.* https://www.gdsinstruments.com/___assets___/pagepdf/000037/Part%5C%201%5C%20Introduction%5C%20to%5C%20triaxial%5C%20testing.pdf (accessed: 11.06.2021)

Rezania, M. (2008). Evolutionary polynomial regression based constitutive modelling and incorporation in finite element analysis. *School of Engineering, Computing and Mathematics, University of Exeter, PhD Thesis.*

Rogers, L., & Dowla, F. (1994). Optimization of groundwater remediation using artificial neural networks with parallel solute transport modeling. *Water Resources Research, 30(2)*, 457–481. https://doi.org/10.1029/93WR01494

Savvidou, C., & Britto, A. (1995). Numerical and experimental investigation of thermally induced effects in saturated clay. *Soils and Foundations, 35*(1), 37–44. https://doi.org/https://doi.org/10.3208/sandf1972.35.37

Selig, E., Gorman, C., Hopkins, T., Deen, R., & Drnevich, V. (1978). Constant-rate-of-strain and controlled-gradient consolidation testing. *Geotechnical Testing Journal, 1(1)*, 3. https://doi.org/10.1520/gtj10363j

Shahin, M. (2016). State-of-the-art review of some artificial intelligence applications in pile foundations. *Geoscience Frontiers, 7(1)*, 33–44. https://doi.org/10.1016/j.gsf.2014.10.002

Shahin, M., Maier, H., & Jaksa, M. (2004). Data division for developing neural networks applied to geotechnical engineering. *Journal of Computing in Civil Engineering, 18(2).* https://doi.org/10.1061/(ASCE)0887-3801(2004)18:2(105)

Shahin, M., Maier, H., & Jaksa, M. (2008). State of the art of artificial neural networks in geotechnical engineering. *The Electronic Journal of Geotechnical Engineering, 8*, 1–26. http://hdl.handle.net/20.500.11937/459407

Shapiro, S., & Yamamuro, J. (2003). Effects of silt on threedimensional stress–strain behaviour of loose sand. *Journal of Geotechnical and Geoenvironmental Engineering, 129(1)*, 1–11.

Simoni, A., & Houlsby, G. (2006). The direct shear strength and dilatancy of sand–gravel mixtures. *Geotechnical and Geological Engineering, 24*, 523–549. https://doi.org/10.1007/s10706-004-5832-6

Smith, I. M., Griffiths, D. V., & Margetts, L. (2015a). Material non-linearity. *Programming the finite element method* (pp. 233–331). John Wiley; Sons, Ltd. https://doi.org/10.1002/9781119189237.ch6

Smith, I. M., Griffiths, D. V., & Margetts, L. (2015b). Static equilibrium of linear elastic solids. *Programming the finite element method* (pp. 169–232). John Wiley; Sons, Ltd. https://doi.org/10.1002/9781119189237.ch5

Smyrniou, E. (2018). Soil constitutive modelling using neural networks. *Faculty of Civil Engineering and Geosciences, Delft University of Technology, M.Sc Thesis.*

Sola, J., & Sevilla, J. (1997). Importance of input data normalization for the application of neural networks to complex industrial problems. *Nuclear Science, IEEE Transactions*, *44*, 1464–1468. https://doi.org/10.1109/23.589532

Sture, S., & Desai, C. (1979). Fluid cushion truly triaxial or multiaxial testing device. *Geotechnical Testing Journal 2*, *1*, 20–33. https://doi.org/10.1520/GTJ10585J

Sussillo, D., & Abbott, L. (2014). Random walk initialization for training very deep feedforward networks. https://www.researchgate.net/publication/269935401_Random_Walks_Training_Very_Deep_Nonlinear_Feed-Forward_Networks_with_Smart_Initialization

Tansel, B., & Guoqing, Z. (2021). *Journal of civil and environmental engineering.* https://www.hilarispublisher.com/civil-environmental-engineering/editor-in-chief.html (accessed: 11.06.2021)

Tech, V. (2021). *Introduction to direct simple shear (dss) testing.* https://www.vjtech.co.uk/blog/introduction-to-direct-simple-shear-dss-testing (accessed: 09.06.2021)

Ting, T., & Horgan, C. (1996). Anisotropic elasticity: Theory and applications. *Journal of Applied Mechanics*, *63*(4), 1056. https://doi.org/10.1115/1.2787237

Varadarajan, A. (2002). Constitutive modeling and application of finite element method in geotechnical engineering. *Indian Geotechnical Journal*, *32(1)*. https://pdfs.semanticscholar.org/e3a3/73f78f202a35c93d7a0c3e1576d9f993ff40.pdf

Xie, S., Lawniczak, A. T., & Hao, J. (2020). Modelling autonomous agents' decisions in learning to cross a cellular automaton-based highway via artificial neural networks. *Computation*, *8*(3). https://doi.org/10.3390/computation8030064

Yashima, A., Leroueil, S., Oka, F., & Guntoro, I. (1998). Modelling temperature and strain rate dependent behavior of clays: One dimensional consolidation. *Soils and Foundations*, *38*(2), 63–73. https://doi.org/https://doi.org/10.3208/sandf.38.2_63

Yin, J.-H., Zhou, W.-H., Kumruzzaman, M., & Cheng, C.-M. (2010). New mixed boundary true triaxial loading device for testing study on 3-d stress-strain-strength behaviour of geomaterials. *Yantu Gongcheng Xuebao/Chinese Journal of Geotechnical Engineering*, *32*, 493–499.

Yu, H.-S. (2006). Perfect plasticity. *Plasticity and geotechnics* (pp. 69–85). Springer US. https://doi.org/10.1007/978-0-387-33599-5_5

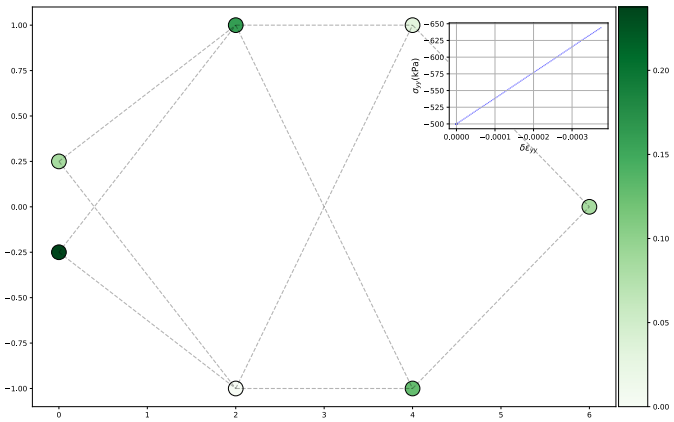
Zhu, J., Zaman, M., & Anderson, S. (1998). Modeling of soil behavior with a recurrent neural network. *Canadian Geotechnical Journal*, *35(5)*. https://doi.org/10.1139/t98-042

Appendices

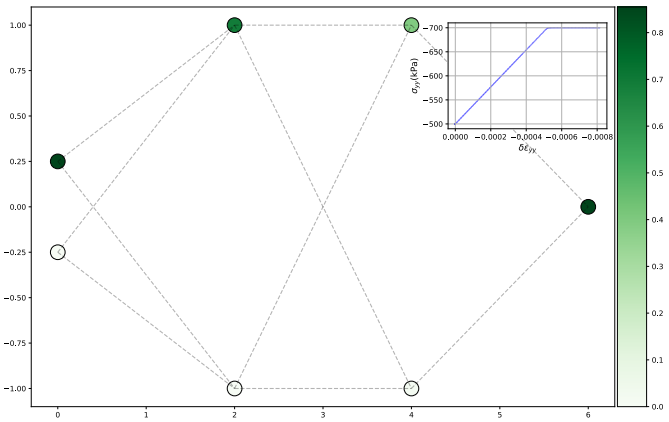
Appendix A

Additional plots and tables

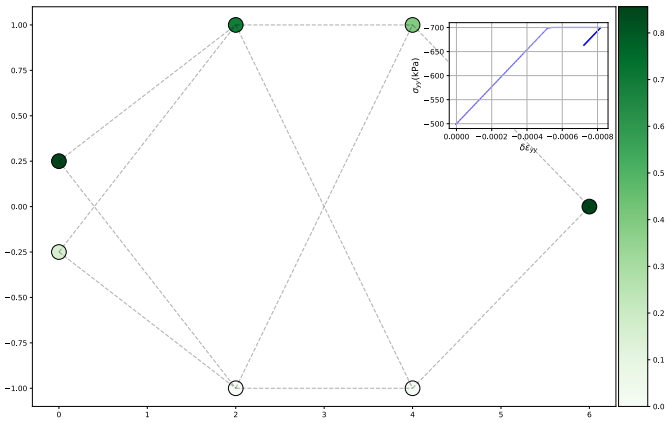
A.1 Understanding Mohr Coulomb ANN by tracking neurons for elastic and plastic states



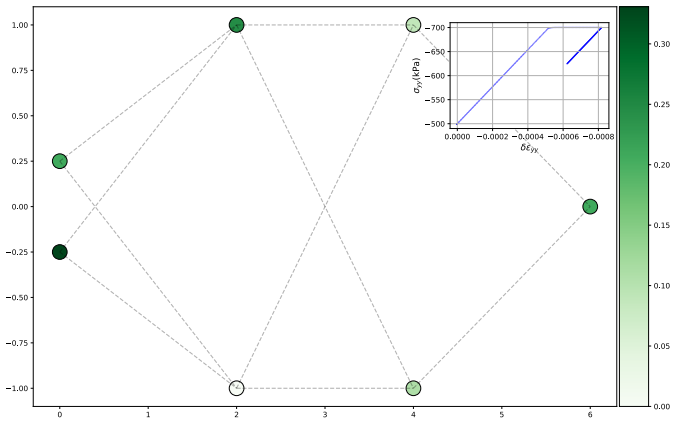
(a) Behaviour of neurons for elastic loading



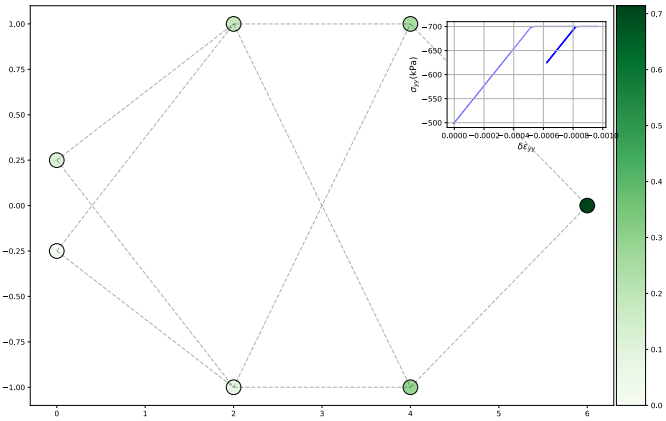
(b) Behaviour of neurons for plastic loading



(c) Behaviour of neurons for elastic unloading



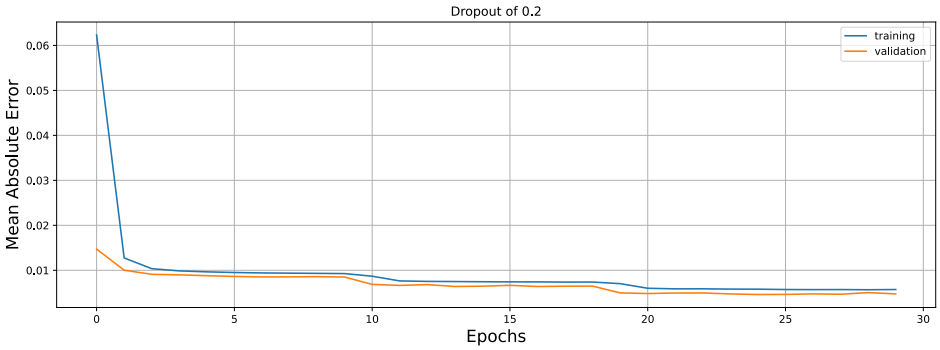
(d) Behaviour of neurons for elastic reloading



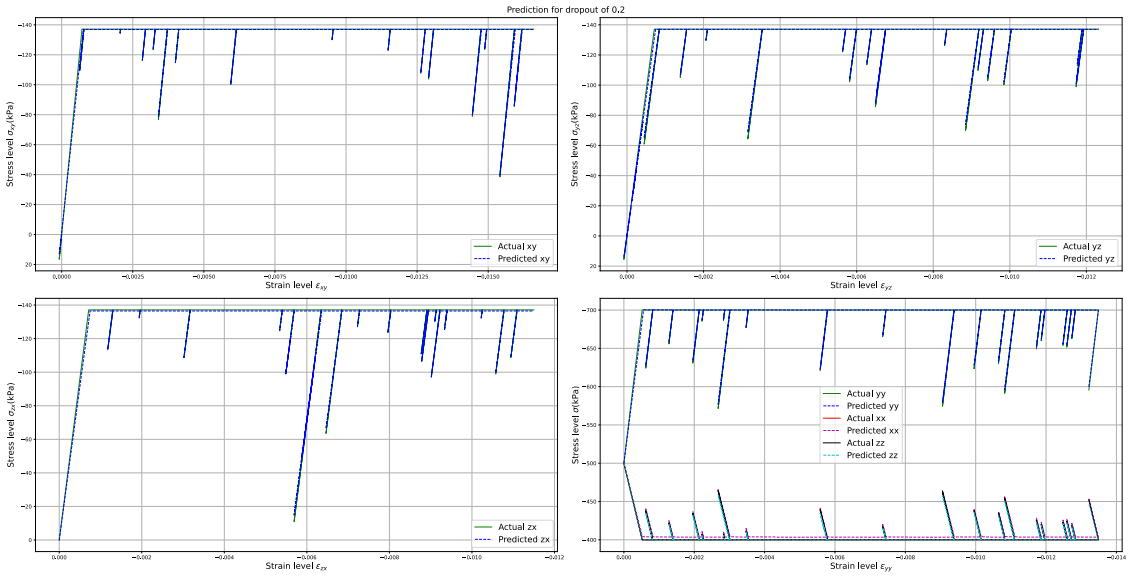
(e) Behaviour of neurons for plastic loading

Figure A.1: Tracking the behaviour of neurons for various soil behaviours

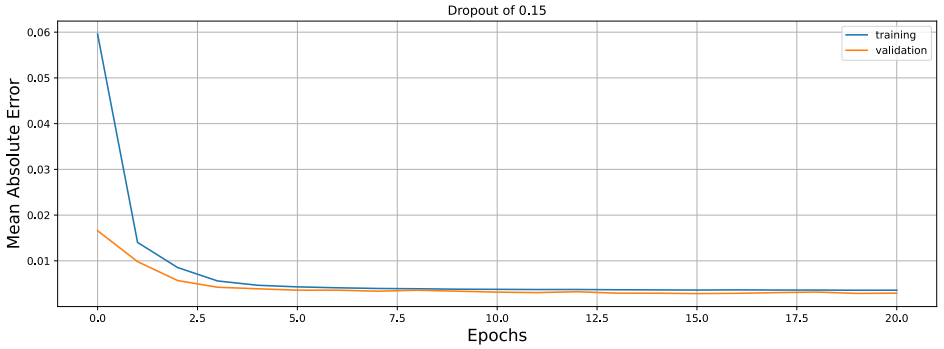
A.2 Predictions for dropouts in Mohr-Coulomb neural network model



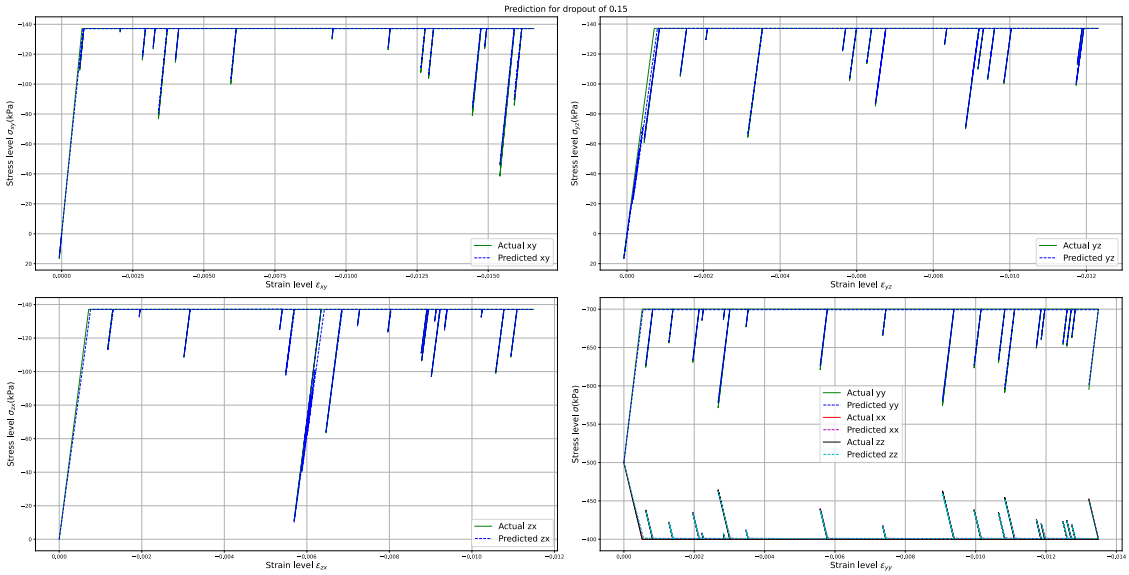
(a) Error in training and validation for dropout of 0.2



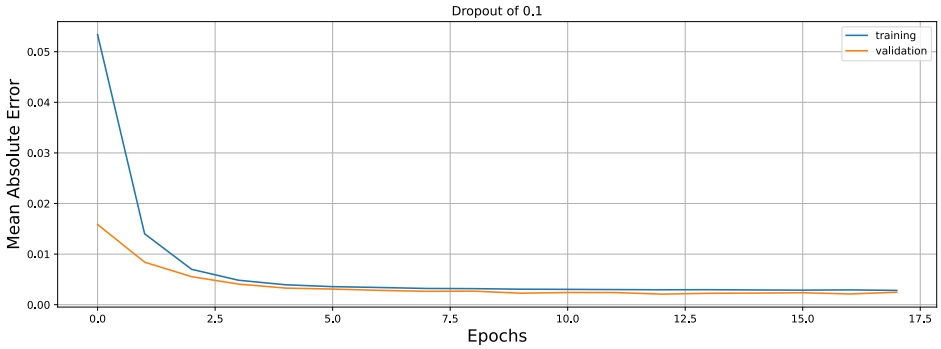
(b) Prediction of network for dropout of 0.2



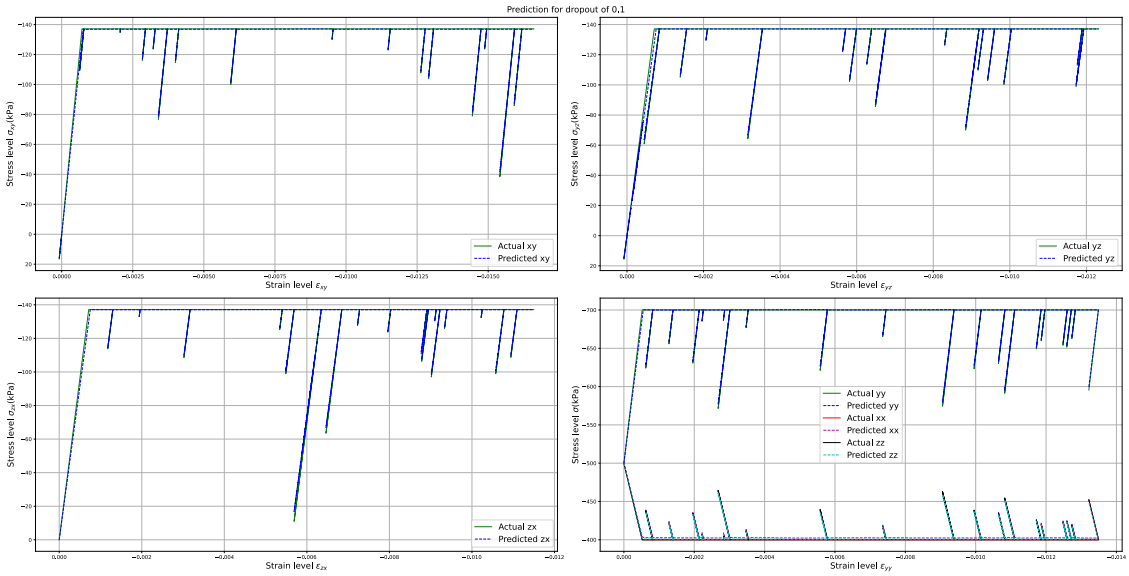
(c) Error in training and validation for dropout of 0.15



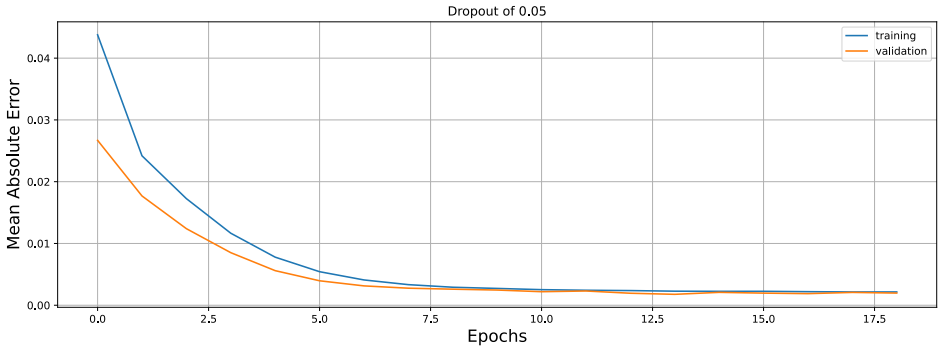
(d) Prediction of network for dropout of 0.15



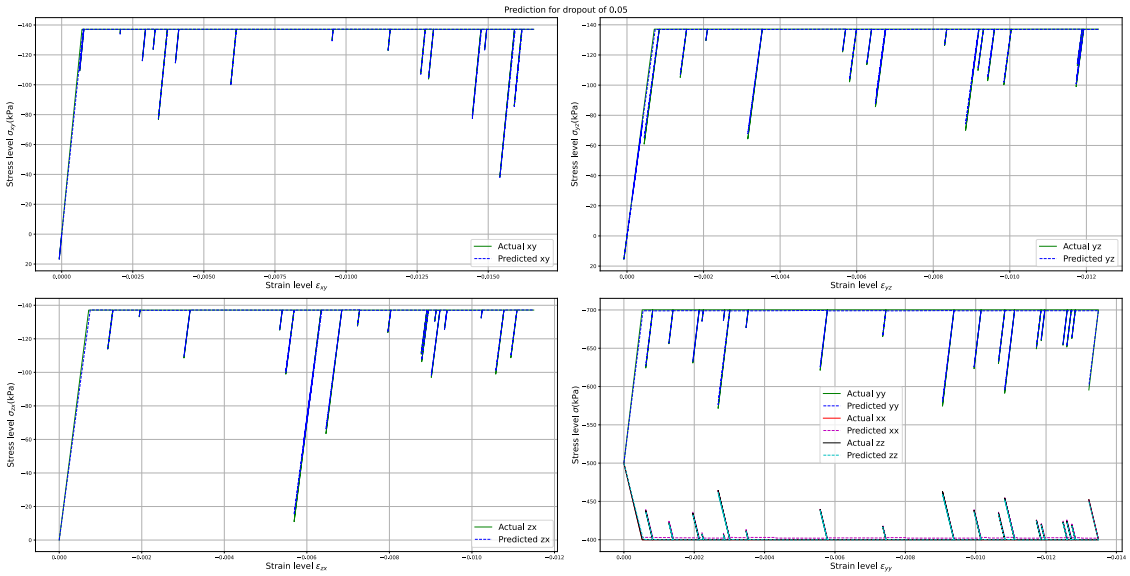
(e) Error in training and validation for dropout of 0.10



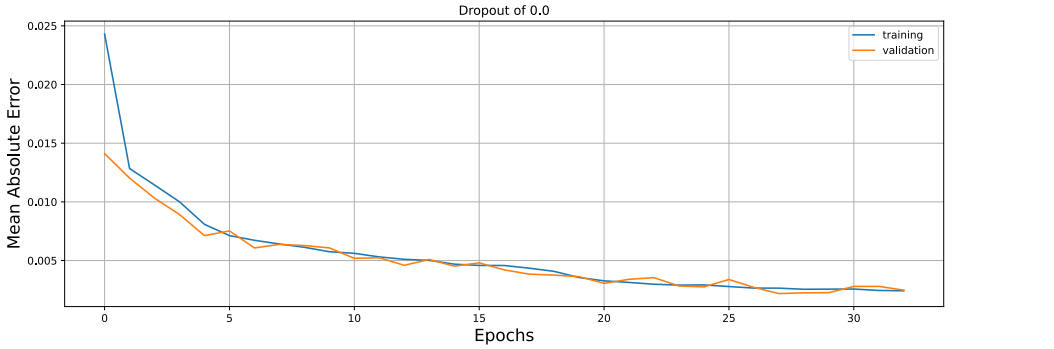
(f) Prediction of network for dropout of 0.10



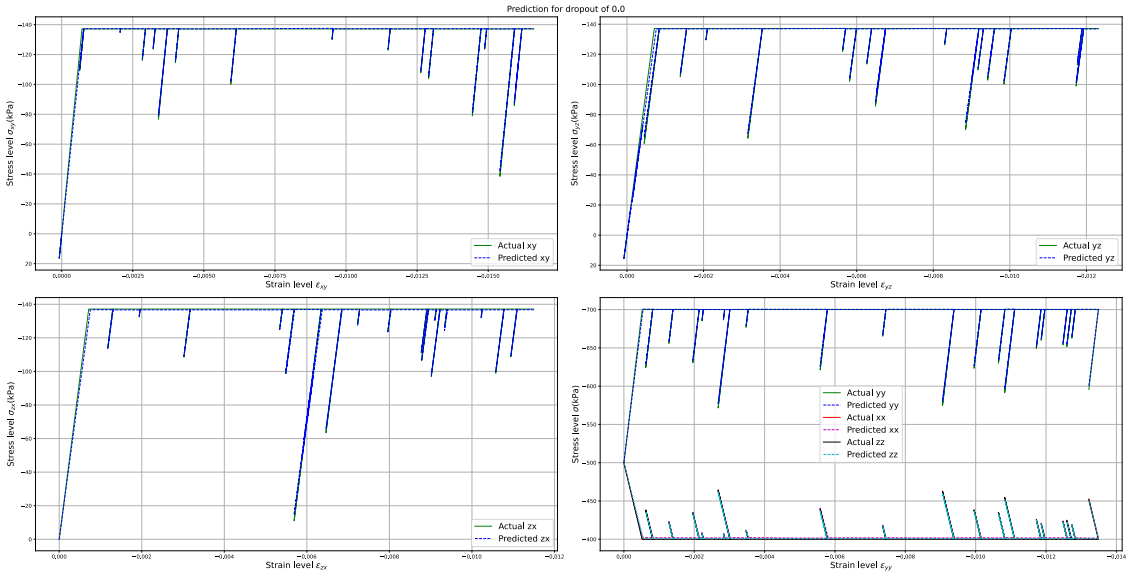
(g) Error in training and validation for dropout of 0.05



(h) Prediction of network for dropout of 0.05



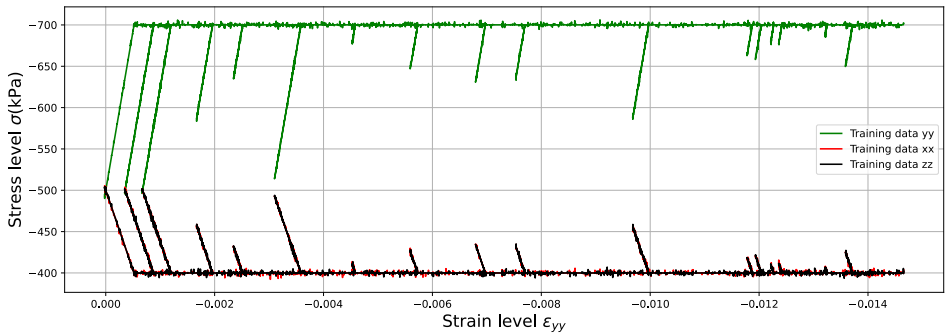
(i) Error in training and validation for dropout of 0.0



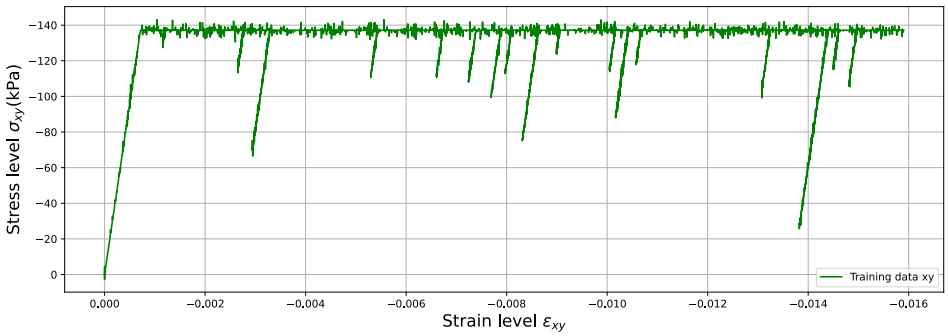
(j) Prediction of network for dropout of 0.0

Figure A.2: Accuracy of Neural network model for various dropouts

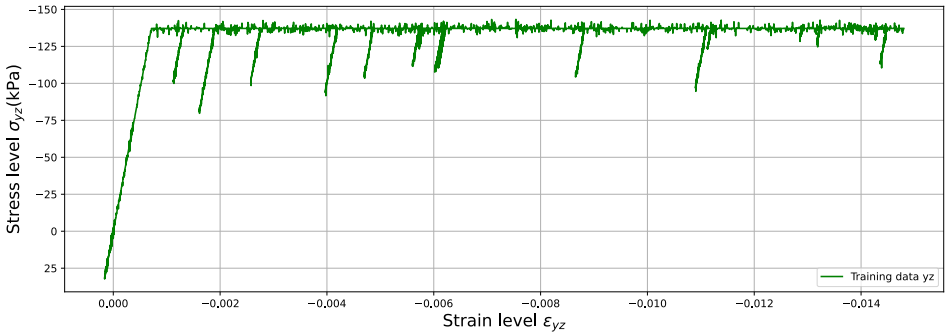
A.3 Sample of training datasets with various noise



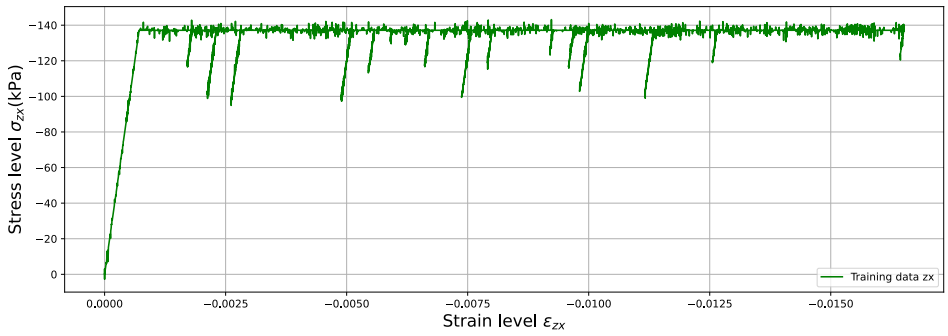
(a) Sample True triaxial training dataset for noise of $\pm 6kPa$



(b) Sample DSS(xy) training dataset for noise of $\pm 6kPa$

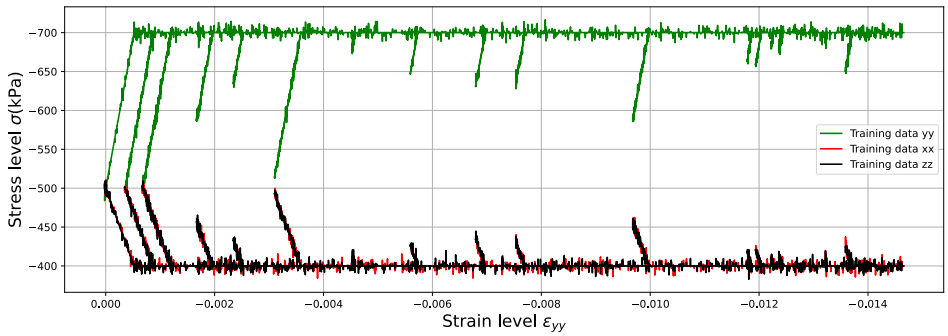


(c) Sample DSS(yz) training dataset for noise of $\pm 6kPa$

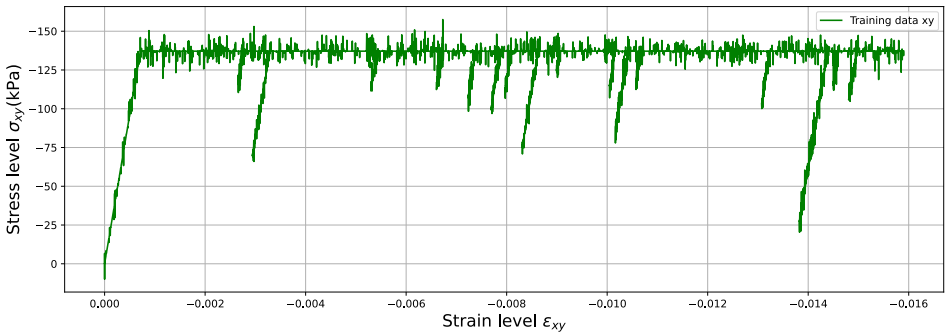


(d) Sample DSS(zx) training dataset for noise of $\pm 6kPa$

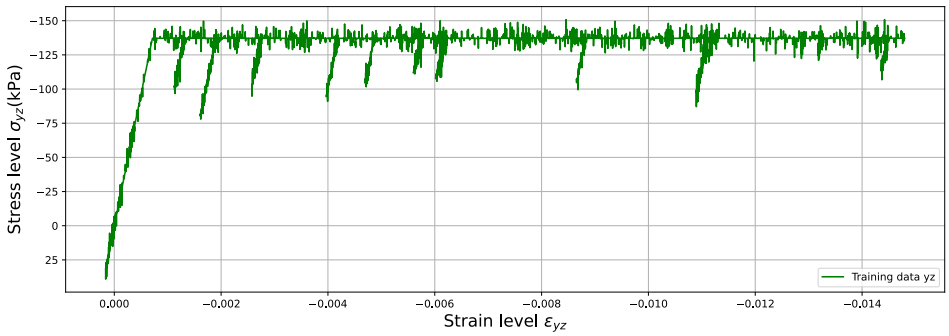
Figure A.3: Sample training datasets for a noise of $\pm 6kPa$



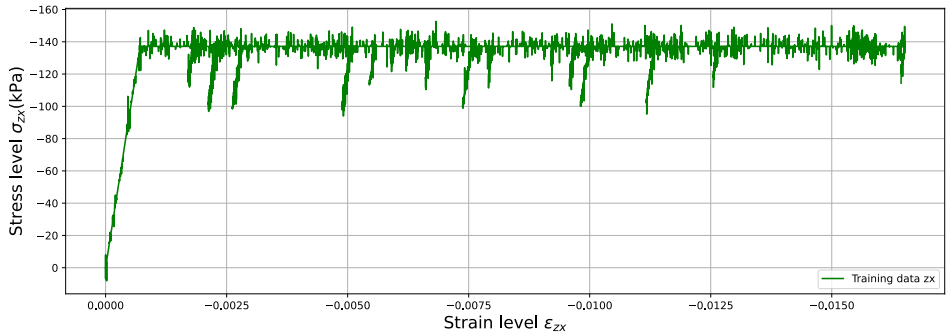
(a) Sample True triaxial training dataset for noise of $\pm 15kPa$



(b) Sample DSS(xy) training dataset for noise of $\pm 15kPa$



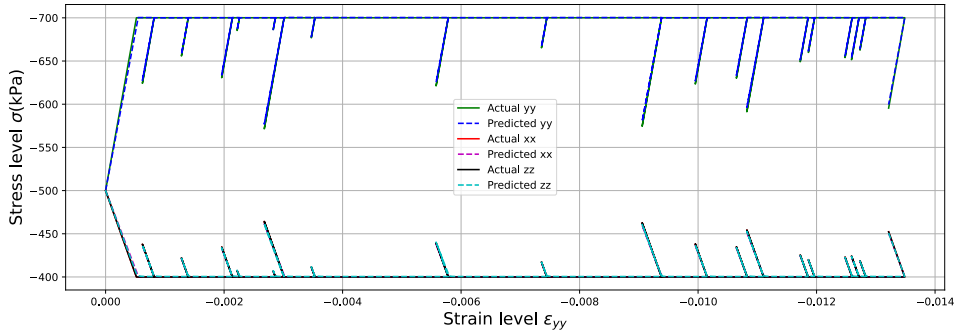
(c) Sample DSS(yz) training dataset for noise of $\pm 15kPa$



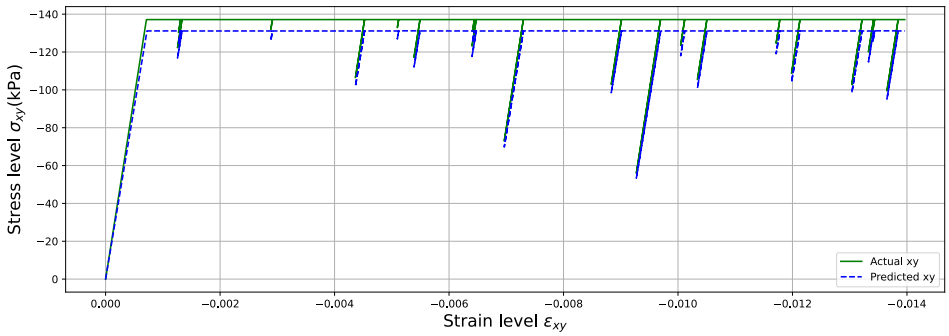
(d) Sample DSS(zx) training dataset for noise of $\pm 15kPa$

Figure A.4: Sample training datasets for a noise of $\pm 15kPa$

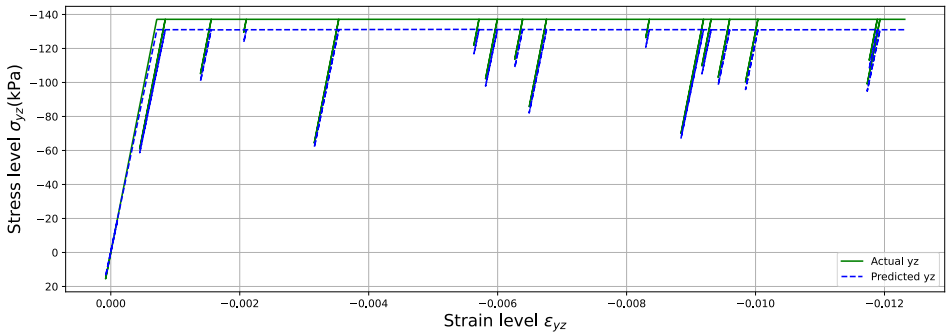
A.4 Prediction when the stress is initialized to -6 kPa in shear directions



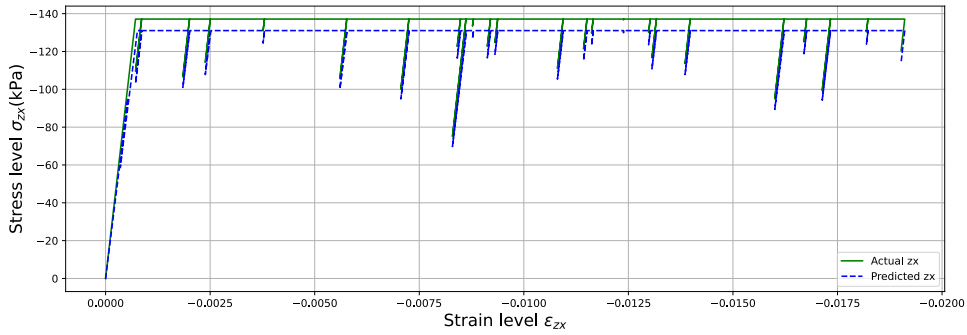
(a) Prediction of stress in principal directions for initial stress of $-6kPa$ in shear directions



(b) Prediction of stress in shear direction xy for initial stress of $-6kPa$ in shear directions



(c) Prediction of stress in shear direction yz for initial stress of $-6kPa$ in shear directions



(d) Prediction of stress in shear direction zx for initial stress of $-6kPa$ in shear directions

Figure A.5: Prediction of Neural network model tested for stress initialized to $-6kPa$ in shear directions

Stress Initialization	Absolute Error					
	$\delta\sigma_{xx}$	$\delta\sigma_{yy}$	$\delta\sigma_{zz}$	$\delta\sigma_{xy}$	$\delta\sigma_{yz}$	$\delta\sigma_{zx}$
Initial stress	$4.962 * 10^{-2}$	$5.020 * 10^{-2}$	$4.966 * 10^{-2}$	$3.883 * 10^{-5}$	$4.277 * 10^{-5}$	$2.435 * 10^{-4}$
of -497 kPa	$4.973 * 10^{-2}$	$4.951 * 10^{-2}$	$4.992 * 10^{-2}$	$1.567 * 10^{-6}$	$4.277 * 10^{-5}$	$2.435 * 10^{-4}$
in principal	$4.973 * 10^{-2}$	$4.951 * 10^{-2}$	$4.992 * 10^{-2}$	$3.883 * 10^{-5}$	$3.566 * 10^{-6}$	$2.435 * 10^{-4}$
directions	$4.973 * 10^{-2}$	$4.951 * 10^{-2}$	$4.992 * 10^{-2}$	$3.883 * 10^{-5}$	$4.277 * 10^{-5}$	$9.184 * 10^{-6}$
Initial stress	$5.031 * 10^{-2}$	$5.090 * 10^{-2}$	$5.035 * 10^{-2}$	$3.883 * 10^{-5}$	$4.277 * 10^{-5}$	$2.435 * 10^{-4}$
of -490 kPa	$4.982 * 10^{-2}$	$4.951 * 10^{-2}$	$4.994 * 10^{-2}$	$1.567 * 10^{-6}$	$4.277 * 10^{-5}$	$2.435 * 10^{-4}$
in principal	$4.998 * 10^{-2}$	$4.951 * 10^{-2}$	$4.992 * 10^{-2}$	$3.883 * 10^{-5}$	$3.566 * 10^{-6}$	$2.435 * 10^{-4}$
directions	$4.973 * 10^{-2}$	$4.951 * 10^{-2}$	$4.996 * 10^{-2}$	$3.883 * 10^{-5}$	$4.277 * 10^{-5}$	$9.184 * 10^{-6}$
Initial stress	$4.929 * 10^{-2}$	$4.999 * 10^{-2}$	$4.93 * 10^{-2}$	$1.333 * 10^{-4}$	$1.574 * 10^{-4}$	$2.683 * 10^{-4}$
of -3 kPa in	$4.933 * 10^{-2}$	$4.95 * 10^{-2}$	$4.98 * 10^{-2}$	$2.899 * 10^{-4}$	$1.574 * 10^{-4}$	$2.697 * 10^{-4}$
shear	$4.933 * 10^{-2}$	$4.95 * 10^{-2}$	$4.98 * 10^{-2}$	$1.334 * 10^{-4}$	$3.076 * 10^{-4}$	$2.694 * 10^{-4}$
directions	$4.933 * 10^{-2}$	$4.95 * 10^{-2}$	$4.98 * 10^{-2}$	$1.334 * 10^{-4}$	$1.574 * 10^{-4}$	$2.932 * 10^{-4}$
Initial stress	$4.929 * 10^{-2}$	$4.999 * 10^{-2}$	$4.93 * 10^{-2}$	$1.333 * 10^{-4}$	$1.554 * 10^{-4}$	$3.590 * 10^{-4}$
of -6 kPa in	$4.933 * 10^{-2}$	$4.95 * 10^{-2}$	$4.98 * 10^{-2}$	$5.87 * 10^{-4}$	$1.574 * 10^{-4}$	$2.981 * 10^{-4}$
shear	$4.933 * 10^{-2}$	$4.95 * 10^{-2}$	$4.98 * 10^{-2}$	$1.334 * 10^{-4}$	$6.047 * 10^{-4}$	$3.192 * 10^{-4}$
directions	$4.933 * 10^{-2}$	$4.95 * 10^{-2}$	$4.98 * 10^{-2}$	$1.334 * 10^{-4}$	$1.573 * 10^{-4}$	$5.902 * 10^{-4}$

Table A.1: Absolute error(in kPa) for various stress initializations

Appendix B

Background information on Relative contribution metric

Relative contribution metric will be explained in this Appendix. The objective of this metric is to quantify the relationship modelled in the validation of neural network rather than using an error measure as the only basis of assessment(Smyrniou, 2018). This gives an estimation on the control of weights of Neural network on the inputs and outputs of the network. Therefore, the contribution of each input can be determined based on the direction and magnitude of the weights. The Overall connection weight is used in this case to determine the contribution of a certain input to the outputOlden and Jackson, 2002. In order to understand this metric, an example is considered as follows:

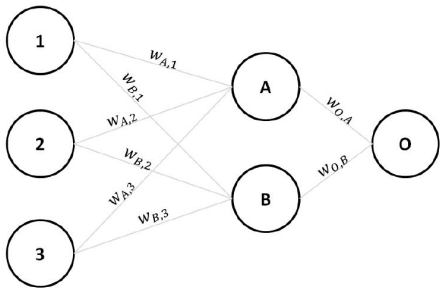


Figure B.1: An Example neural network considered to compute RC(Smyrniou, 2018)

In order to compute the relative contribution of the above network, overall connection weights(OCWs) are required. The contribution of input 1 can be obtained by initially computing the values of $c_{A,1}$, $c_{B,1}$ which represent the contribution through hidden node A and B respectively. Using the following equations, OCW is computed:

$$c_{A,1} = c_{A,1} * c_{O,A} \tag{B.1}$$

$$c_{B,1} = c_{B,1} * c_{O,B} \tag{B.2}$$

$$OCW_1 = c_{A,1} + c_{B,1} \tag{B.3}$$

OCW obtained is an approximation of the real relationship of input(Smyrniou, 2018). This metric does not take into account the efect of activation function or the biases of the network. However, in Olden and Jackson, 2002, it was found that the connection weight approach was efficiently able to model the importance of each input in the neural network. The Relative Contribution(RC) for each input can be computed as follows:

$$RC_1 = \frac{OCW_1}{|OCW_1| + |OCW_2| + |OCW_3|} \tag{B.4}$$

Similarly,

$$RC_2 = \frac{OCW_2}{|OCW_1| + |OCW_2| + |OCW_3|} \tag{B.5}$$

$$RC_3 = \frac{OCW_3}{|OCW_1| + |OCW_2| + |OCW_3|} \tag{B.6}$$