

CSE 397: Final Project

A Neural Network Model for Hyperelasticity

Soovadeep Bakshi, William Ruy,
Akshay Kumar Varanasi, Wenbo Zhang

Prof. Tan Bui-Thanh

Fall 2018

Abstract

This project is about building a neural network as a surrogate constitutive model for computational solid mechanics. Traditional phenomenological models often have simple forms based on physical insights, but they lack ability to account for complex material behaviors. Although meso-scale/multi-scale material models are very predictive, most of them are computational expensive when complex mechanical behaviors are involved, such as heterogeneous nonlinear materials. This gives rise to the necessity for a surrogate model that has the ability to represent such responses. To this end, we will investigate possible approaches to build a neural network model that satisfies the rules for the hyperelastic model.

1 Mathematical Model

We consider the strain energy density function of a hyperelastic model, $\Psi = \Psi(\mathbf{C})$, as a function of right Green-Lagrangian deformation tensor $\mathbf{E} \in \mathbb{S}_{++}^2$, where \mathbb{S}_{++}^2 is the set of all symmetric matrix in $\mathbb{R}^{2 \times 2}$. To avoid duplicated terms, we use Voigt notation to concatenate the independent components of \mathbf{E} into a vector $\mathbf{E} \in \mathbb{R}^3$. Then, we have the strain energy density Φ and the second Piola-Kirchhoff stress \mathbf{S} ,

$$\Psi = \Psi(\mathbf{E}, \mathbf{X}), \quad (1)$$

$$\mathbf{S} = \frac{\partial \Psi}{\partial \mathbf{E}}, \quad (2)$$

where \mathbf{X} is the parameters for the model. We denote $\Psi_x = \Psi(\mathbf{E} | \mathbf{X} = \mathbf{x})$ as the strain energy density function for fixed parameters \mathbf{x} .

The hyperelastic model we are interested in is the structural model of the native or cross-linked soft tissue [3]. It is called structural model in the sense

that it is developed by taking into account the meso-structure of the material. The material is a composite of matrix and collagen fibers. It is predictive but computationally expensive due to the fact that it involves integrals over the collagen fiber architecture.

The structural model for native tissues has two contributions from the matrix and the collagen fibers,

$$\Psi_{nat} = \Psi_{mat} + \Psi_{col}. \quad (3)$$

The matrix term is a modified Yeoh model,

$$\Psi_{mat} = \phi_{mat} \frac{\eta_M}{2} \left(\frac{1}{a} (I_1 - 3)^a + \frac{r}{b} (I_1 - 3)^b \right) \quad (4)$$

where, ϕ_{mat} is the mass fraction of matrix, η_M is the modulus parameter, I_1 is the first invariant of the right Cauchy–Green deformation tensor $\mathbf{C} = 2\mathbf{E} + \mathbf{I}$, a, b are the shape parameters, and r is the weight between the two terms. The parameters are required to satisfy the inequalities, $1 < a < b$, $ab < 2$, and $r \geq 0$.

The collagen term is an ensemble average over the fiber orientation distribution function (ODF), Γ , and the recruitment distribution function (RDF), D ,

$$\Psi_{col} = \phi_{col} \eta_C \int_{\theta} \Gamma(\theta) \int_1^{\lambda_{\theta}} D(\lambda_{\theta}) \left(\frac{\lambda_{\theta}}{\lambda_s} - 1 \right)^2 d\lambda_{\theta} d\theta. \quad (5)$$

where, ϕ_{col} is the mass fraction of matrix, η_C is the modulus of the collagen fibers, $\lambda_{\theta} = \sqrt{\theta \cdot \mathbf{C} \cdot \theta}$ is the stretch in θ direction, λ_s is the slack stretch, $\lambda_{\theta}/\lambda_s$ is the true stretch after collagen fibers are straightened.

The structural model for cross-linked tissues has three contributions from the matrix, the collagen fibers, and the fiber-fiber interaction,

$$\Psi_{cro} = \Psi_{mat} + \Psi_{col} + \Psi_{int}. \quad (6)$$

The interaction term, Ψ_{int} , is more computational expensive since it is a quadruple integral. To accurately compute the quadruple integral with tensor product quadrature rule, it requires $21^4 = 194481$ quadrature nodes. This give rise to the need of a computationally efficient surrogate model that can replicate the same behavior as the structural model while respect the fundamental principles for the hyperelastic constitutive model, i.e. material frame indifference, material symmetry, stability/convexity conditions.

Non-negativity

To be physically meaningful, Ψ needs to be non-negative everywhere.

$$\Psi \geq 0 \quad \forall \mathbf{E} \quad (7)$$

$$\Psi = 0 \text{ iff } \mathbf{E} = \mathbf{0} \quad (8)$$

Convexity

Denote tangential stiffness matrix $\mathbf{C} \in \mathbb{R}^{3 \times 3}$ as

$$\mathbf{C} = \frac{\partial^2 \Psi}{\partial \mathbf{E}^2}. \quad (9)$$

The local convexity condition for Ψ requires that

$$\mathbf{C}(\mathbf{E}) \succ 0 \quad \forall \mathbf{E}. \quad (10)$$

Another direct result is that \mathbf{C} is symmetric since the energy density function is continuously differentiable. So \mathbf{C} is symmetric and positive definite everywhere,

$$\mathbf{C}(\mathbf{E}) \in \mathbb{S}_{++}^3 \quad \forall \mathbf{E}. \quad (11)$$

where \mathbb{S}_{++}^3 is the set of all symmetric positive semi-definite matrix in $\mathbb{R}^{3 \times 3}$.

Material Frame Indifference

Material frame indifference requires that the formula is objective under rigid body motion of the current configuration. As long as we are using variables that are defined on the reference configuration, this is not a concern for our application. As \mathbf{E} is defined on the reference configuration, so we will use \mathbf{E} as the measure of strains.

Inputs & Outputs

The strain energy function of a hyperelastic model can have different mathematically equivalent forms. Selecting the appropriate inputs and outputs carefully would be beneficial for imposing those fundamental principles. Usually, the data from physical models or experiments are strains and their corresponding stresses. So the training set for the surrogate model contains the stress-strain data.

Training and validation data

The data set for training and validation are tuples of strains and its corresponding stresses, $D = \{(\mathbf{E}^{(i)}, \mathbf{S}^{(i)})\}_{i=1}^n$. We can restrict our attention to a bounded range of strains, defined as component-wise inequality $\mathbf{E}_{lb} \leq \mathbf{E} \leq \mathbf{E}_{ub}$, instead of the whole space \mathbb{R}^3 , because large strains/stresses are of little interest for real-world applications. Denote the index set as I_{tr} for the training part, and I_{val} for the validation part.

The training data set is composed of linear spaced $11 \times 11 \times 11$ samples for $\mathbf{E}_{lb} = (0, 0, -0.15)$ and $\mathbf{E}_{ub} = (0.17, 0.23, 0.15)$, and the validation data set includes $25 \times 25 \times 25$ for $\mathbf{E}_{lb} = (0, 0, -0.15)$ and $\mathbf{E}_{ub} = (0.17, 0.23, 0.15)$.

For plotting the results, we use three subsets of the validation data: $25 \times 25 \times 1$ for $\mathbf{E}_{lb} = (0, 0, 0)$ and $\mathbf{E}_{ub} = (0.17, 0.23, 0)$, $1 \times 25 \times 25$ samples for $\mathbf{E}_{lb} = (0, 0, -0.15)$ and $\mathbf{E}_{ub} = (0.0, 0.23, 0.15)$, $25 \times 1 \times 25$ for $\mathbf{E}_{lb} = (0, 0, -0.15)$ and $\mathbf{E}_{ub} = (0.17, 0.0, 0.15)$.

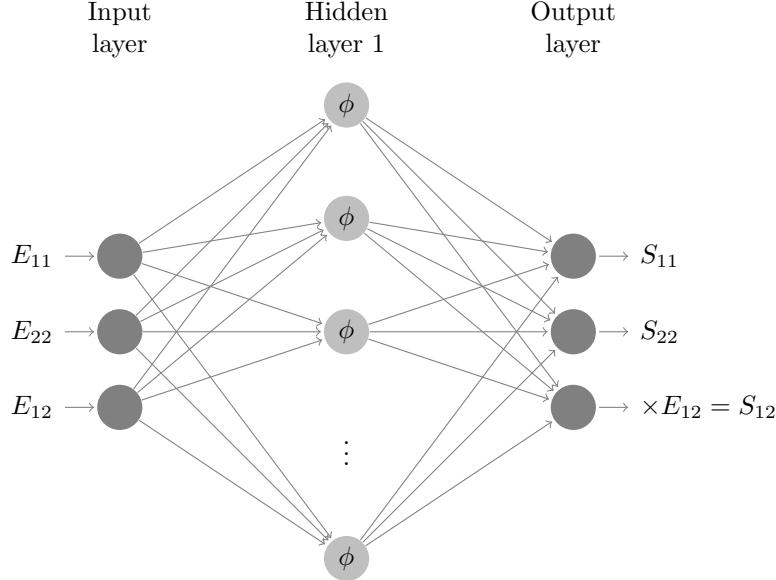


Figure 1: The neural network for strain-to-stress map.

2 Neural Network Models

2.1 Strain-to-stress map

Since the data set is composed of strains and their corresponding stresses, i.e., $D = \{(\mathbf{E}^{(i)}, \mathbf{S}^{(i)})\}_{i=1}^n$, It would be natural to feed the strains into the neural network and get stresses from it, which is illustrated in Figure 1.

$$\mathbf{E} \implies \text{Neural Network} \implies \mathbf{S}$$

Then, this becomes a function approximation problem. Since the stress-to-strain map is continuous and its support is restricted to a compact set in \mathbb{R}^3 , it is feasible to use the multi-layer perceptron for function approximation by the universal approximation theorem.

$$\text{find} \quad \tilde{\mathbf{S}}, \quad (12a)$$

$$\text{minimize} \quad \frac{1}{|I_{\text{tr}}|} \sum_{i \in I_{\text{tr}}} (\tilde{\mathbf{S}}(\mathbf{E}^{(i)}) - \mathbf{S}^{(i)}). \quad (12b)$$

We used one hidden layer with 21 neurons with the sigmoid function as activation function. We plotted the validation result in Figure 2 for equi-biaxial test ($E_1 = E_2, E_3 = 0$).

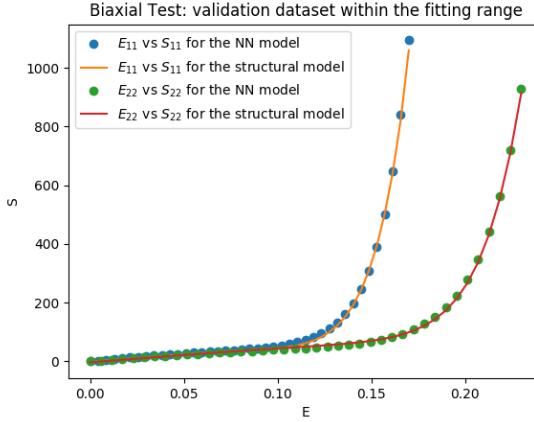


Figure 2: The validation result of the fitted neural network for biaxial test ($E_1 = \frac{0.23}{0.17}E_2$, $E_3 = 0$).

We used Adam [1] for training. Even though Adam does not guarantee convergence for convex optimization problem [2], we find the algorithm converges for our cases in this section. Since the function is continuous, we use the maximum norm of the discrepancy between $\tilde{\mathbf{S}}(\mathbf{E}^{(i)})$ and $\mathbf{S}^{(i)}$ for all i ,

$$\max_{i \in I_{val}} \|\tilde{\mathbf{S}}(\mathbf{E}^{(i)}) - \mathbf{S}^{(i)}\|_{\infty}, \quad (13)$$

to calibrate the validation error. The fitted neural network model is compared with the equi-biaxial validation test in Figure 2. It shows very good fitting for this loading path.

In Figure ??, we list the fitting results in comparison with validation data for $E_{11} - E_{22}$ plane, $E_{22} - E_{12}$ plane and $E_{11} - E_{12}$ plane. As it shows, all the eight cases except Figure ??(g) gives good fitting results. The validation error for the trained neural network is 92.8646 in this case. This is because the shear stress S_{12} oscillates when the shear strain E_{12} is zero.

This can be solved by slightly altering the neural network. We multiply the third output by the shear strain E_{12} , and refit the data again. The refitting results are given in Figure 3. The validation error for this neural network is reduced to 27.6851. As shown in Figure 3(g), the error is comparable with the rounding error.

Thus, we have a very good fitting for strain-stress map with only one layer of 21 neurons. Even though this architecture fits the strain-to-stress map very well, one concern from the pure mechanistic perspective is that the tangential stiffness is not symmetric in general, i.e. $\mathbf{C} \neq \mathbf{C}^T$. This can be fixed by fitting the strain-to-energy map instead, but the problem left is: Can we fit the data very well?

9

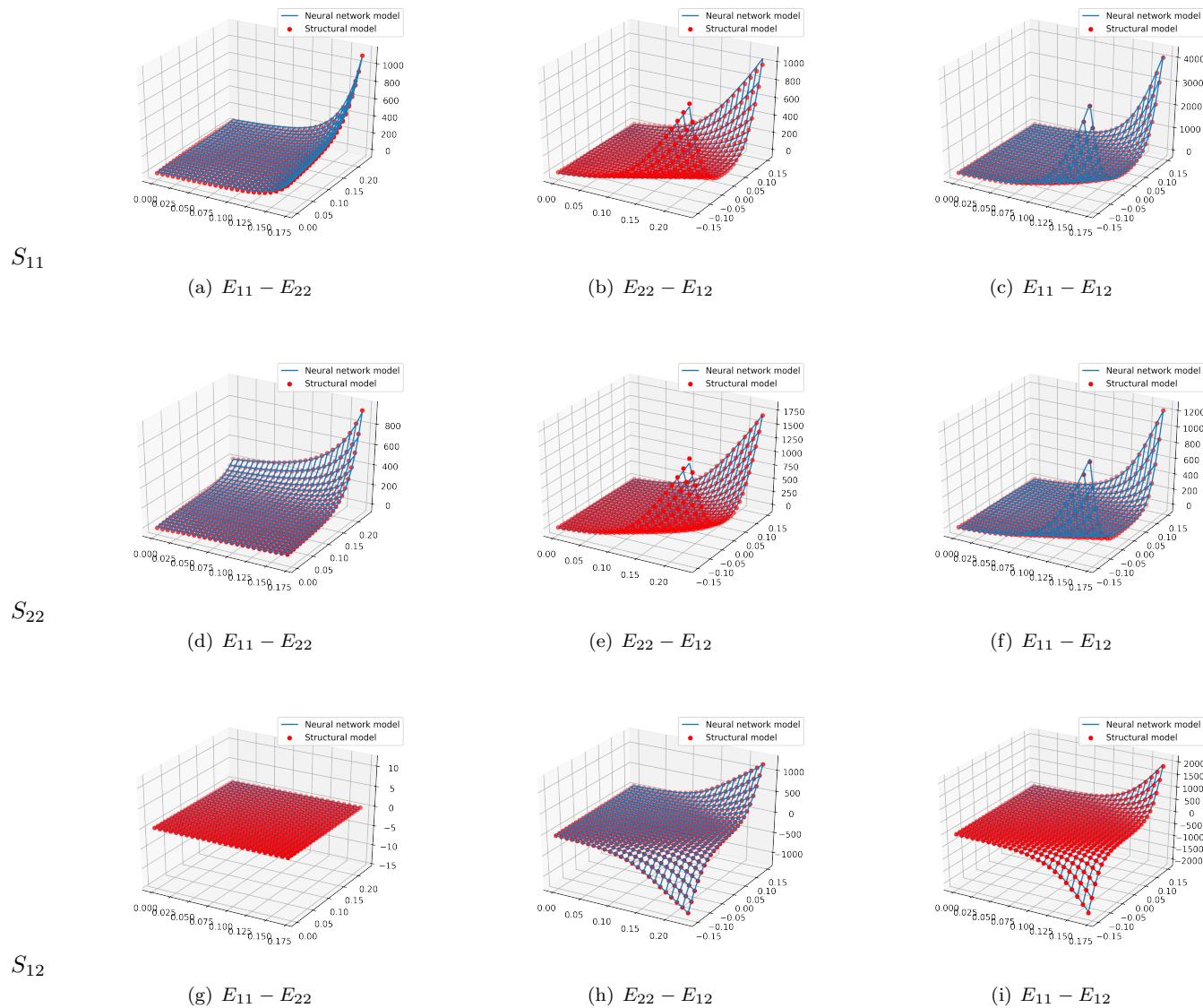


Figure 3: Validation results for strain-stress map - 2

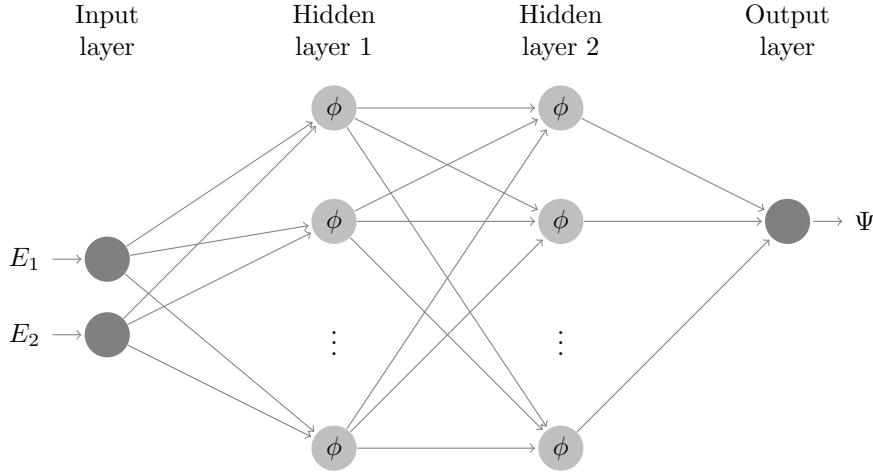


Figure 4: The neural network for strain-to-energy map described in Section 2.2.1.

2.2 Attempts to fit the strain-to-energy map

In this section, we did not include E_3 (E_{12}) since the learning algorithm did not converge when we include it. This may be due to the fact that we are trying to match the gradient of the estimated strain energy density function with the stress data, since the neural network is now a strain-to-energy map instead. However, this needs to be studied further and will be examined in future work. We want to find a $\tilde{\Psi}$ such that its gradients fit the stress data.

$$\text{find} \quad \tilde{\Psi}, \quad (14a)$$

$$\text{minimize} \quad \sum_{i \in I_{\text{tr}}} \left(\frac{\partial \tilde{\Psi}}{\partial \mathbf{E}}(\mathbf{E}^{(i)}) - \mathbf{S}^{(i)} \right). \quad (14b)$$

2.2.1 Sigmoid function as activation function

We also used two hidden layers, each of which has ten neurons with the sigmoid function as activation function, as shown in Figure 4. We plotted the validation result in Figure 5 for equi-biaxial test ($E_1 = E_2, E_3 = 0$). We can see the neural network overfitted the data.

2.2.2 Smoothed maximum over squares

To overcome the overfitting problem, we first recall some important characteristics of convex functions and convex sets.

Lemma 2.1 $\Psi(\mathbf{E})$ is a convex function of \mathbf{E} if and only if its epigraph $\text{epi}(\Psi) = \{(\mathbf{E}, t) \mid t \geq \Psi(\mathbf{E})\}$, is a convex set.

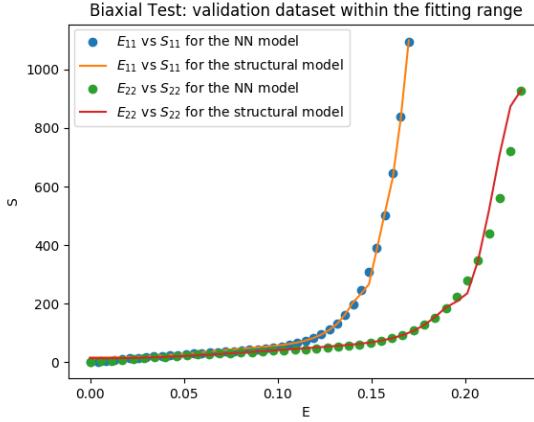


Figure 5: The biaxial ($E_1 = \frac{0.23}{0.17}E_2$, $E_3 = 0$) validation result of the fitted neural network described in Section 2.2.1.

Lemma 2.2 *Any closed convex set C can be written as an intersection of a possibly infinite set of halfplanes.*

By Lemma 2.1 and 2.2, Ψ can be approximated by a maximum function over a possibly infinite set of linear functions. Since Ψ has non-zero Hessian everywhere and its Hessian plays an important role as the tangential stiffness matrix, we cannot use piecewise linear function. Otherwise, the Hessian would be zero almost everywhere and the Ψ is discontinuous unless infinite number of linear functions were used. Then, this is not a efficient way to approximate Ψ .

Notice that the strain energy function for collagen fiber in Equation 5, Ψ_{col} , is a weighted sum of quadratic functions of stretch in the fiber directions. Then Ψ_{col} must be α -strong convex β -strong smooth with respect to $\mathbf{C}^{1/2}$ for some α and β . On the other hand, by the physical meaning of the recruitment function, more fibers are recruited as strains increase. This phenomenon is very similar to the behavior of the rectified linear unit (ReLU). Even though ReLU is not differentiable, the square of ReLU is actually differentiable. An heuristic intuition would lead to using a sequence of gradually growing quadratic function as the strains grow. α would be the leading coefficient of the quadratic function at origin, β would be that at the edge of the range of strains.

We then heuristically use the hypothesis space composed of the form of function as:

$$\Psi = \eta + \lambda \log \sum_i \left\{ \exp \left[\sum_j \mathbf{C}_i [\text{ReLU}^2(\mathbf{A}_j \mathbf{E} + \mathbf{b}_j)] + \mathbf{d}_i \right]^2 \right\} \quad (15)$$

where log-sum-exp is a smoothed version of maximum function.

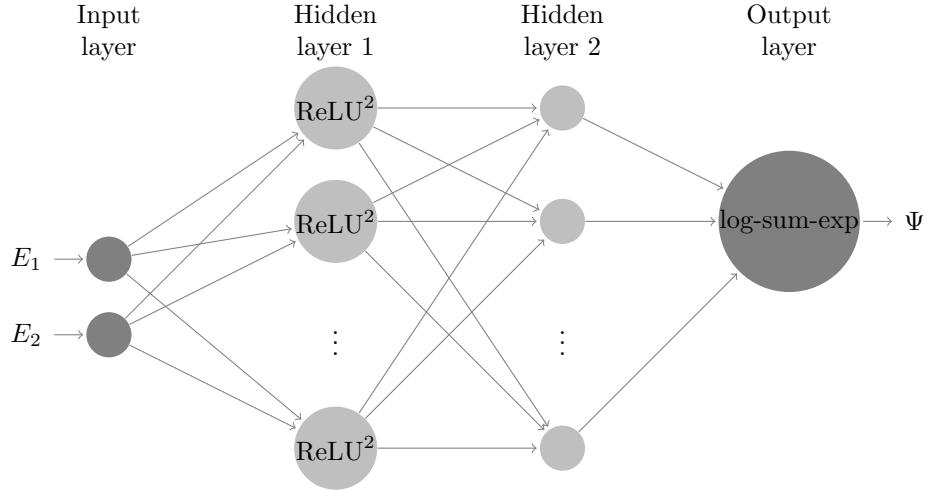


Figure 6: The neural network for strain-to-energy map described in Section 2.2.2.

We plotted the validation result in Figure 7 for equi-biaxial test ($E_1 = E_2$, $E_3 = 0$). It is shown that the overfitting problem is solved by this approach.

3 Conclusion and Future Work

In this project, we have looked at building a neural network as a surrogate constitutive model for computational solid mechanics. We show results for two different models to fit the data, i.e., the strain-to-stress mapping and the strain-to-energy mapping, and both models show reasonable convergence results for both models. Therefore, this project serves as a proof-of-concept for the use of neural networks in replacing constitutive models. However, issues with both the methods have been pointed out, and have to be examined in future work.

Currently, the working example in Section 2.2.2 does not take into account the shear E_3 (E_{12}). When we feed the shear E_3 into the neural network, the training does not converge, possibly due to the fact that we are trying to fit the derivative of the strain energy function with the stresses for the training data. Next, we may try to come up with other approaches to overcome these deficiencies.

References

- [1] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

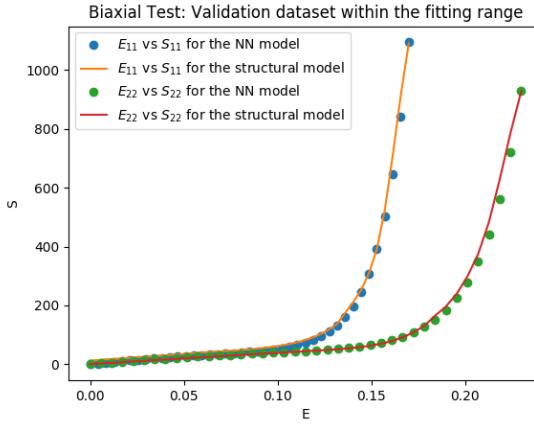


Figure 7: The biaxial validation result of the fitted neural network described in Section 2.2.2 ($E_1 = \frac{0.23}{0.17} E_2$, $E_3 = 0$).

- [2] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. 2018.
- [3] Michael S Sacks, Will Zhang, and Silvia Wognum. A novel fibre-ensemble level constitutive model for exogenous cross-linked collagenous tissues. *Interface focus*, 6(1):20150090, 2016.