Towards neural network models for describing the large deformation behavior of sheet metal

Maysam B. Gorji 1*, Dirk Mohr 2

1 Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA
2 Department of Mechanical and Process Engineering, ETH Zurich, Tannenstrasse 3, 8092, Zurich, Switzerland

* gorji@mit.edu

Abstract. Neural networks provide a potentially viable alternative to a differential equation based constitutive models. Here, a neural network model is developed to describe the large deformation response of a Levy-von Mises sheet material with isotropic strain hardening. Using a conventional return-mapping scheme, virtual experiments are performed to generate stress-strain data for random monotonic biaxial loading paths (up to strains of 0.2). Subsequently, a basic feedforward neural network model is trained and validated using the results from virtual experiments. The results for a shallow network with only two hidden layers show remarkably good agreement with all experimental data. The identified neural network model is implemented into a user material subroutine and used in basic structural simulations such as uniaxial tensile and notched tension experiments. In addition to demonstrating the potential of neural networks for modeling the rate-independent plasticity of metals, their application to more complex problems involving strain-rate and temperature effects is discussed.

Keywords. Plasticity, machine learning, constitutive model, finite elements.

1. Introduction
Proper identifying a material model in finite element analysis – to predict the strain and stress distribution and consequently failure and fracture – is often an extensive and tedious task when a complex material model needs to be studied. Mechanical characterization of material is mainly determined through the high number of experiments and applying the hybrid experimental-numerical approach [1-5]; it is emerging to employ a fully automated testing system (that will be soon available) to perform experiments. In practice, by employing the robot-assisted automated testing systems we have access to a large dataset. In light of the Big Data revolution and Open Data movement, this issue has recently received attention in a variety of applications ranging from engineering to management science. In this paper, we show the machine learning (here the neural network) as an alternative method for a differential equation based constitutive model. The high potential of machine-learning based models for describing the large deformation response of engineering materials at different temperature and strain rate is elaborated in Section 3. In Section 4, the neural network approximation of plane stress plasticity for non-proportional loading paths is discussed.
2. Artificial Neural Networks

Loosely analogous to biological neural networks, the computing system known as an artificial neural network (NN) is introduced in order to solve problems in a similar way that a brain works. A feedforward neural network is the most common architecture that is used in this research; the unidirectional signals flow from the first layer to the latest one by passing through all neurons in the system. The typical NN architecture composes of input layer (to bring the initial data into the system), hidden layer (interconnected nodes that are described through the assigned weights and biases), and output layer (to produce the outputs of the system). Two fundamental factors to model an NN are:

I) The synapse of a neuron described as weight and bias. Without regards to the input data in a given pattern, the bias nodes are always on by considering them connected to a predefined input with value 1. This node increases the flexibility of the model and helps to produce output even when entire feature values are set to zero. The value of the weight and bias is considered as the strength of the connection between the layers.

II) An activation function in order to control the amplitude of the neuron’s output. The output signal of each neuron is transformed by the activation function \( g \) of the weighted sum of the whole input signals \( x_i \) as \( g(\sum_{i=1}^{n} w_i \cdot x_i + b) \). “\( w_i \)” and “\( b \)” denote weight vectors of inputs and bias value of a neuron, respectively. In its simplest form, this function behaves in a binary manner. However, in order to produce a non-linear decision boundary, the non-linearity needs to be added via appropriate non-linear activation functions. The logistic function and especially the S-shape sigmoid function is one the most popular activation functions that imply the non-linearity into the network,

\[
g(X) = \frac{1}{1 + e^{-X}}
\]

where \( e \) is the Euler’s number and \( X \) is referring to the linear combination of the predictors and the weights that are referenced in the question. In turn, the sigmoid function allows modeling responses that vary non-linearly with its explanatory variables; it takes any inputs from the \(- \infty \) to \(+ \infty \), but it confines them to output between range 0 and 1 when the denominator gets so large or so small, respectively. The output of the hidden layer is the non-linear sigmoid activation function \( g(X) \) of a linear combination of inputs. Using linear algebra and dot products of the weights and input vectors \( l \) can be summarized as a single mathematical equation \( H_i = g(b_i^{(l)} + \sum w_{ij}^{(l)} \cdot x_i) \); \( i \) is the layer number. Ultimately, the activation function between the hidden \( (H) \) and the output layer \( (O) \) is prescribed as a just weighted sum \( O_i = b_i^{(l)} + H_i \sum w_{ij}^{(l)} \).

The learning capability of the network is achieved by adjusting all weights and biases. To measure the accuracy of the network, the training model is targeting minimization of the so-called objective function (e.g. mean squared error) with respect to the weights and biases of the network,

\[
J(\Omega) = \sum_{i=1}^{M} \left( y_i - F(x_i;\Omega) \right)^2
\]

where \( \Omega \) stands for all adjustable parameters (weights and biases) and the \( y_i \) is the desired output responses contained in the training set composed of \( M \) data-set. \( F(x_i;\Omega) \) is the calculated output of the network.

3. NN Approximation of Temperature and Strain-rate Dependent Hardening

Mechanical characterization of material at elevated temperature and high strain rate using the physical-based is usually very challenging and computationally expensive to develop; it needs to be incorporated
with strain, strain rate hardening and temperature as well as taking into account the softening effect of dynamic recovery and recrystallization. It requires a highly-complex model to represent all the mentioned parameters. Some of the well-known models are Johnson-Cook [6], Zener-Hollomon [7], Zerilli–Armstrong [8], Durrenberger et al. [9] and Abed and Makarem [10]. In this section, it is discussed how the NN-based hardening model can describe the flow stress at different boundary conditions.

3.1. Virtual Experiments made by Zerilli–Armstrong Model

The well-known Zerilli–Armstrong constitutive relations are widely used to characterize the hardening of material at different temperature and strain rate hardening. This function is modified by Mirzaie et al. [11] to better represent the thermal softening of carbon steel. Here, we use this modified version of the Zerilli–Armstrong to generate the virtual experiments as a function of temperature (T) and strain rate (\(\dot{\varepsilon}\)),

\[
\sigma = 6771.175\varepsilon^{0.232} \exp\left(-\beta_0 T + 0.00012T \cdot \dot{\varepsilon}\right)
\]

\[
\beta_0 = 0.0241\varepsilon^3 - 0.036\varepsilon^3 + 0.0183\varepsilon^3 - 0.0027\varepsilon + 0.0029
\]

Accordingly, 105 virtual experiments were generated to cover the full spectrum of strain rate (15 different loading conditions from quasi-static 0.001/s to extreme dynamic 1000/s loading conditions) and temperature (7 different thermal conditions from room temperature to extreme hot condition 1250°C). Fig. 1 shows the stress-strain relation of fourteen virtual experiments at two different strain rates (\(\dot{\varepsilon} = 0.001/s\) and \(\dot{\varepsilon} = 1000/s\)); it concludes both hardening and softening part of the flow curve.

![Stress-strain relation of virtual experiments at two different strain rates](image)

**Figure 1.** Generated virtual experiments at two different strain rates.

3.2. Machine Learning based Temperature and Strain-Rate Hardening

Creating the machine learning-based constitutive model to imitate the behavior of the equivalent strain-stress is explained in this subsection; the input layer consists of three-neuron representing the plastic strain \(\varepsilon_p[-]\), temperature \(T[K]\), and strain rate \(\dot{\varepsilon} [1/s]\). One neuron specifies the actual true stress \(\sigma [Pa]\) for the outer layer. The non-linearity has been added into the NN system via the logistic sigmoid function. The results show that a single- and double-hidden layer NN model including up to 20 neurons cannot provide satisfactory approximation accuracy in the range of the training strain domain. Figure 2 shows the NN approximation of Zerilli–Armstrong hardening law for 3 random temperature (333°C, 888°C, and 1111°C) and 4 different random strain rates (0.002, 3, 82, and 900/s). Networks with three-hidden layer provide satisfactory approximation – with less than 1% – error by using at least 15 neurons.
Figure 2. NN approximation of strain rate and temperature dependent hardening curve. Black solid and dashed red lines represent virtual experiments and NN approximation at 0.002, 3, 82, and 900/s.

4. NN Approximation of Plane Stress Plasticity for Non-proportional Loading Paths

4.1. Virtual Experiments made by Conventional J$_2$ Plasticity Model

As the second example, the potential of NN to predict the plasticity of material is challenged. Here, the term virtual experiments are used to make reference to numerical simulations with the conventional J$_2$ plasticity model of DP780 [12]. We focus on strain-driven virtual experiments, i.e. a loading path is prescribed as a time history in the strain space \( \{\varepsilon_{11}, \varepsilon_{22}, \varepsilon_{12}\} \) (model input), while the corresponding stresses \( \{\sigma_{11}, \sigma_{22}, \sigma_{12}\} \) are calculated (model output). A series of synthetic stress-strain trajectories were numerically generated with the FE-code ABAQUS (comprises a four-node plane stress element with the in-plane dimensions of 1x1mm) subjected to the proportional and non-proportional loadings. The generated deformations in this part are inspired by the strain evolutions in the metal forming applications, which are developing in the uniform linear elongation at the beginning of the stretching and gradually drifting towards a plane strain path before a fracture occurs [13,14]. This trend is formulated in two consecutive steps:

\[
\varepsilon_i = R \cdot \varepsilon, \quad \varepsilon \leq \varepsilon_0
\]

\[
\varepsilon_i = R \cdot \left\{ \varepsilon_0 + \frac{1}{C_0} \left[ 1 - e^{-C_0(\varepsilon - \varepsilon_i)} \right] \right\}, \quad \varepsilon \geq \varepsilon_0
\]
where \( R \) is denoting the strain ratio between the uniaxial compression \((R = -2)\) and the equi-biaxial tension \((R = 1)\) loadings. \( \varepsilon_0 \) and \( C_0 \) are two auxiliary variables defining the bifurcation strain from the linear path and modifying the shape of the curvature, respectively. The major strain \( \varepsilon_I \) data point will be taken along the path using the power law function of time grid points, which is adjusted with different exponents,

\[
\varepsilon_I = C_2 \left( \bar{\varepsilon} \right)^{C_1}, \quad \bar{\varepsilon} \leq C_1
\]
\[
\varepsilon_I = C_2 \cdot \left( C_1 + \frac{1}{C_4} \left[ 1 - e^{-C_1(\varepsilon - \varepsilon_0)} \right] \right)^{\varepsilon_0}, \quad \bar{\varepsilon} \geq C_1
\]  

(5)

Seven considered adjustable parameters \( R, \varepsilon_0, \) and \( C_0\) assist playing with the strain evolution in both normal-shear and principle strains space. Three-dimensional input strain vector is then simply calculated using the rotational matrix in different angles \( \theta \):

\[
\begin{bmatrix} \varepsilon_{11} & \varepsilon_{12} \\ \varepsilon_{12} & \varepsilon_{22} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} \varepsilon_1 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \end{bmatrix}^T \begin{bmatrix} \varepsilon_1 \\ 0 \end{bmatrix} = \begin{bmatrix} \varepsilon_1 \\ 0 \end{bmatrix} \end{bmatrix}
\]  

(6)

Figure 3. Training data set under the non-linear loading strain paths in \( \{ \varepsilon_I, \varepsilon_{11} \} \) domain.

Figure 3 shows the nonlinear input strain evolution that is generated from strain paths in \( \{ \varepsilon_I, \varepsilon_{11} \} \) domain. The total chosen trajectories in \( \varepsilon_I - \varepsilon_{11} \) space are seven that are transformed in various angles i.e., the total training set data are 840 paths. The input vector is initially rewritten in \( \varepsilon_I - \varepsilon_{11} \) space and then transferred to the three-dimensional input variable space \( \{ \varepsilon_{11}, \varepsilon_{22}, \varepsilon_{12} \} \). They will serve as a time like parameter in the simulations for non-proportional loading.

4.2. Machine Learning based Constitutive Model

This section is dedicated to create a machine learning based constitutive model and teach the neural network to imitate the behavior of a material governed by J2 plasticity model. The synthetic experimental data of the single element is employed to train the neural network. The elastic-plastic history of the current strain and applied strain history from the initial stress-free state at \( t = 0 \) to the current deformed state at \( t = \tau \) are specified for the input layer of a network in order to predict the stress tensor at a time \( t = \tau \):

\[
\begin{bmatrix} \text{input} \\ \varepsilon^{(r)}_{11}, \varepsilon^{(r)}_{22}, \varepsilon^{(r)}_{12}, \varepsilon^{(r)}_{11}, \varepsilon^{(r)}_{22}, \varepsilon^{(r)}_{12} \end{bmatrix} \rightarrow \begin{bmatrix} \text{output} \\ \sigma^{(r)}_{11}, \sigma^{(r)}_{22}, \sigma^{(r)}_{12} \end{bmatrix}
\]  

(7)
where

\[
\bar{\varepsilon}_{ij}(\tau) = \frac{1}{\tau} \int_{0}^{\tau} \varepsilon_{ij}(t)dt
\]

\(\bar{\varepsilon}_{ij}(\tau)\) is introduced to take into account the history of strains and it is independent of time.

### 4.3. Model implementation into a finite element code

A number of hidden layers and neurons undeviatingly improve the learning. However, the crucial point that needs to be considered is the efficiency of the system. Choosing complicated architecture is computationally and FE-implémentary expensive. The desired NN architecture that is implemented in our FE code concludes two layers with 30 neurons. Thereupon, the total number of adjustable parameters that can be tweaked and turned to make this network behave differently is 1233 parameters i.e., 1170 scaler weights (6x30 parameters between the input and first hidden layer, 30x30 parameters between two hidden layers, and 3x30 parameters between the last hidden layer and output one) and 63 biases (30, 30, and 3 parameters for the first, second hidden layer, and the output layer, respectively). The initial calculated mean squared error between the output and the target stress is 1.7E+6; the best performance in 1000 complete pass through a given dataset is gained in the latest epoch with the performance of 1.18.

Note that 70% of the generated data is used to train the network and the rest 30% is utilized for the validation and testing of the network. The latter set is fully isolated from the rest of the virtual experiments i.e., the performance of the network (see Eq. 2) is measured accordingly. This operation is constantly accomplishing during and after the training. It is worth mentioning that to improve the ANN generalization, the overfitting of the data should be prevented. To do so, we applied the automated regularization of the normalized Input-Output data set in combination with Levenberg-Marquardt training function [16, 17]; the performance function is then modified by adding the squared weights and biases. Therefore, smaller weights and biases are obtained and consequently, the network response is smoother, which helps to prevent the overfitting.

In the last step, the optimal weights and biases of the network (they are calculated in MATLAB) were exported and compiled into the programming language FORTRAN in order to embed it into user-defined material subroutines (ABAQUS/Explicit VUMATs). The identified NN-based constitutive model, implemented into a user material subroutine, is then used in basic structural simulations. The results were then compared with the von-Mises plasticity model.

### 4.4. Numerical Simulation for Structural Application: \(J_2\) Plasticity vs. NN Constitutive Model

#### 4.4.1. Uniaxial Tensile Test

The results of the uniaxial tensile experiment based on the vM yield function evaluated up to 1.6mm displacement (see the black curve in Fig. 4). The almost same response obtained when the implemented NN-based constitutive model is used (see blue curve in Fig. 4). The computed force by both models increase monolithically up to 5kN at a displacement of about 0.12mm, and it then reaches to maximum force 8kN at a displacement of about 1.5mm.

Figure 4 depicts selected snapshots of the planar stress-contour plots for studied models. Up to the maximum force, a pronounced deformation is homogeneously distributed in the gage section throughout the entire loading history. Using the NN constitutive model produces the same structural response as the J2 model provided (see (1) to (3) in Fig. 4). Indeed, comparison of the axial stress field distributions between the conventional von Mises yield criterion and the NN reveal the potential of the latter approach; there is a perfect agreement between them.
Figure 4. Comparing the performance of the $J_2$ plasticity (generated virtual experiment) with the NN-based constitutive model in a uniaxial tensile test [15].

Notched Tensile Test. Figure 5 summarizes the results for the notched tension specimen. The force maximum of 3.5kN is reached after applying a displacement of 0.45mm; the $J_2$ numerically-predicted force is then decreased. A similar trend is modeled in the trained NN; it exhibits a maximum of 3.8kN at a displacement of 0.61mm. There is a perfect agreement up to 0.16mm displacement between these models. Beyond this point, the NN model slightly overestimates the force prediction with a difference of 4% and 7% at displacement 0.32mm and 0.48mm, respectively. This discrepancy is attained to the accumulation of the predicted stress in every single element of the model. In fact, a bigger range of experimental points to train the network as well as using more hidden layers and neurons can improve the results. It is worth mentioning that the overall calculated stress response of the machine learning technique is seen to be in good agreement with the $J_2$ plasticity model. The strains are localized in the center of the specimen throughout the entire loading history, while the remaining parts are relatively intact. The maximum planar axial stresses of the critical element based on the von Mises yield criterion in steps ①, ②, and ③ are 778MPa, 877MPa, and 930MPa, respectively. The trained NN predicts correspondingly 781MPa, 886MPa, and 941MPa. There is thereupon less than 1% difference; it shows the strength of NN to retain the constitutive of a material.

Figure 5. Comparing the performance of the $J_2$ plasticity (generated virtual experiment) with the NN-based constitutive model in a notched tensile test.

5. Conclusions
In this research, we showed that an artificial neural network is an alternative tool that can assist in modeling a material response and predicting plasticity in metal forming applications. The artificial neural network model has been deployed to describe the plane stress response of a von Mises material with isotropic hardening. In the first step, two different synthetic stress-strain paths have been generated on the 1×1 mm single shell element in the strain space. History of strains $\varepsilon_{11}, \varepsilon_{22}$ and $\varepsilon_{12}$ are defined for the input layer of a neural network to predict the stresses $\sigma_{11}, \sigma_{22}$ and $\sigma_{12}$ as an output layer. The obtained multiaxial responses of the single-element are then modeled by the feedforward neural network; two hidden layers with thirty neurons are employed for processing the data received from the input layer. The derived parameterized neural network is afterward implemented as a user material.
routine for the FE package ABAQUS. The model is then applied on the flat dogbone-shaped and the notched specimens to predict the material responses. There is a great agreement between the stresses predicted by NN model and virtual experiments generated by $J_2$ plasticity model; it shows the potential of machine learning as an alternative approach describing the constitutive model of a material.

Acknowledgements
The partial financial support of the MIT Industrial Fracture Consortium is gratefully acknowledged. Special thanks are due to Prof. Peyman Mohajerin, Dr. Thomas Tancogne-Dejean, and Mr. Wei Li for valuable discussions.

References
[1] Y. Bao, T. Wierzbicki, 2004, On fracture locus in the equivalent strain and stress triaxiality space, Int. J. Mech. Sci. 46 (1) 81-98.
[2] Erice, B., Pérez-Martín, M.J. and Gálvez, F., 2014. An experimental and numerical study of ductile failure under quasi-static and impact loadings of Inconel 718 nickel-base superalloy. International Journal of Impact Engineering, 69, pp.11-24.
[3] Gorji MB, Mohr D, 2017, Micro-tension and Micro-Shear experiments to characterize stress-state dependent ductile fracture. Acta Materialia, 131:65-76.
[4] Brüning, M., Gerke, S. and Schmidt, M., 2018. Damage and failure at negative stress triaxialities: Experiments, modeling and numerical simulations. International Journal of Plasticity, 102, pp.70-82.
[5] Gorji MB, Mohr D, 2018, Predicting Shear Fracture of Aluminum 6016-T4 during Deep Drawing: Combining Yld-2000 Plasticity with Hosford-Coulomb Fracture Model. IJMS, 137, pp.105-120.
[6] Johnson and W.H. Cook. Fracture characteristics of three metals subjected to various strains, strain rates, temperatures and pressures. Engineering Fracture Mechanics, 21:31-48, 1985.
[7] Zener, C. and Hollomon, J.H., 1944. Effect of strain rate upon plastic flow of steel. Journal of Applied physics, 15(1), pp.22-32.
[8] Zerilli, F.J. and Armstrong, R.W., 1987. Dislocation-mechanics-based constitutive relations for material dynamics calculations. Journal of applied physics, 61(5), pp.1816-1825.
[9] Durrenberger, L., Klepaczko, J.R., Rusinek, A., 2007. Constitutive Modeling of Metals Based on the Evolution of the Strain-Hardening Rate. J. Eng. Mater. Technol. 129, 550.
[10] Abed, F., Makarem, F., 2012. Comparisons of Constitutive Models for Steel Over a Wide Range of Temperatures and Strain Rates. J. Eng. Mater. Technol. 134, 21001.
[11] Mirzaie, T., Mirzadeh, H. and Cabrera, J.M., 2016. A simple Zerilli–Armstrong constitutive equation for modeling and prediction of hot deformation flow stress of steels. Mechanics of Materials, 94, pp.38-45.
[12] Marcadet SJ, Mohr D (2016) Critical hardening rate model for predicting path-dependent ductile fracture. International Journal of Fracture, 200(1-2), pp.77-98
[13] Gorji, M., Berisha, B., Hora, P. and Barlat, F., 2016. Modeling of localization and fracture phenomena in strain and stress space for sheet metal forming. International Journal of Material Forming, 9(5), pp.573-584.
[14] Gorji, M., Berisha, B., Manopulo, N. and Hora, P., 2016. Effect of through thickness strain distribution on shear fracture hazard and its mitigation by using multilayer aluminum sheets. Journal of Materials Processing Technology, 232, pp.19-33.
[15] Gorji MB, Mohr D (2019) A Basic Neural Network Model Describing the Plasticity of Sheet Metal, submitted to Numiform 2019: The 13th International Conference on Numerical Methods in Industrial Forming Processes, Portsmouth, New Hampshire, USA.
[16] MacKay, D.J.C., “Bayesian interpolation,” Neural Computation, Vol. 4, No. 3, 1992, pp. 415–447.
[17] Foresee, F.D., and M.T. Hagan, “Gauss-Newton approximation to Bayesian regularization,” Proceedings of the 1997 International Joint Conference on Neural Networks, 1997, pp. 1930–1935.