Vol. 03, No. 3,(2019),143–158, **ISSN:2581-3242**

IJMLNCE JOURNAL

International Journal of Machine Learning and Networked Collaborative Engineering

Journal Homepage: http://www.mlnce.net/home/index.html

DOI : https://doi.org/10.30991/IJMLNCE.2019v03i03.003

Understanding Machine Learning for Friction Stir Welding Technology

Akshansh Mishra[*]

Center for Artificial Intelligence and Friction Stir Welding , Stir Research Technologies, UP, India

akshansh.frictionwelding@gmail.com (https://orcid.org/0000-0003-4939-359X)

Abstract

Machine learning process drastically decreases the time it takes to develop stronger, lighter materials. This is important to the aerospace, automotive and manufacturing sectors. Machine Learning techniques like Artificial Neural Networks and Image processing are used in a solid state joining process such as Friction Stir Welding process (FSW) for optimization of mechanical properties like Ultimate Tensile Strength (UTS), Fracture Strength , Elongation percentage and microstructure properties like grain size and understanding defects formation. In the recent paper, application of Machine Learning technique in Friction Stir Welding technology will be discussed.

Keywords

Machine Learning,

Friction Stir Welding,

Neural Network,

Tensorflow,

Artificial Intelligence,

Google Colaboratory

1. Understanding Human Nervous System

The human nervous system can be classified as a three stage system which is shown in the block diagram of Figure 1. The brain is in central position to the nervous system which is represented by the neural net. The function of the neural net is to receive the information continuously, perceive it and then to make appropriate decision.

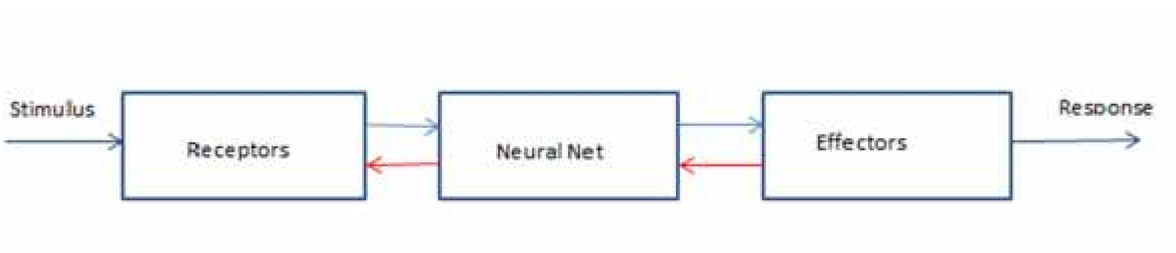


Figure 1. Representation of Nervous System by block diagram

[*] Corresponding author

Akshansh Mishra

Email: akshansh.frictionwelding@gmail.com

The green arrows which are pointing from left to right represent the forward transmission of information carrying signals through the nervous system. The red arrows in the diagram show the presence of feedback in the nervous system. Stimuli from the external environment or the human body are converted to electrical impulses by receptors which further convey the information to the brain (neural net). Electrical impulses which are generated by the receptors are further converted to a distinguishable response as a nervous system outputs by the effectors [1]. Figure 2 represents the schematic diagram of the human nerve cell.

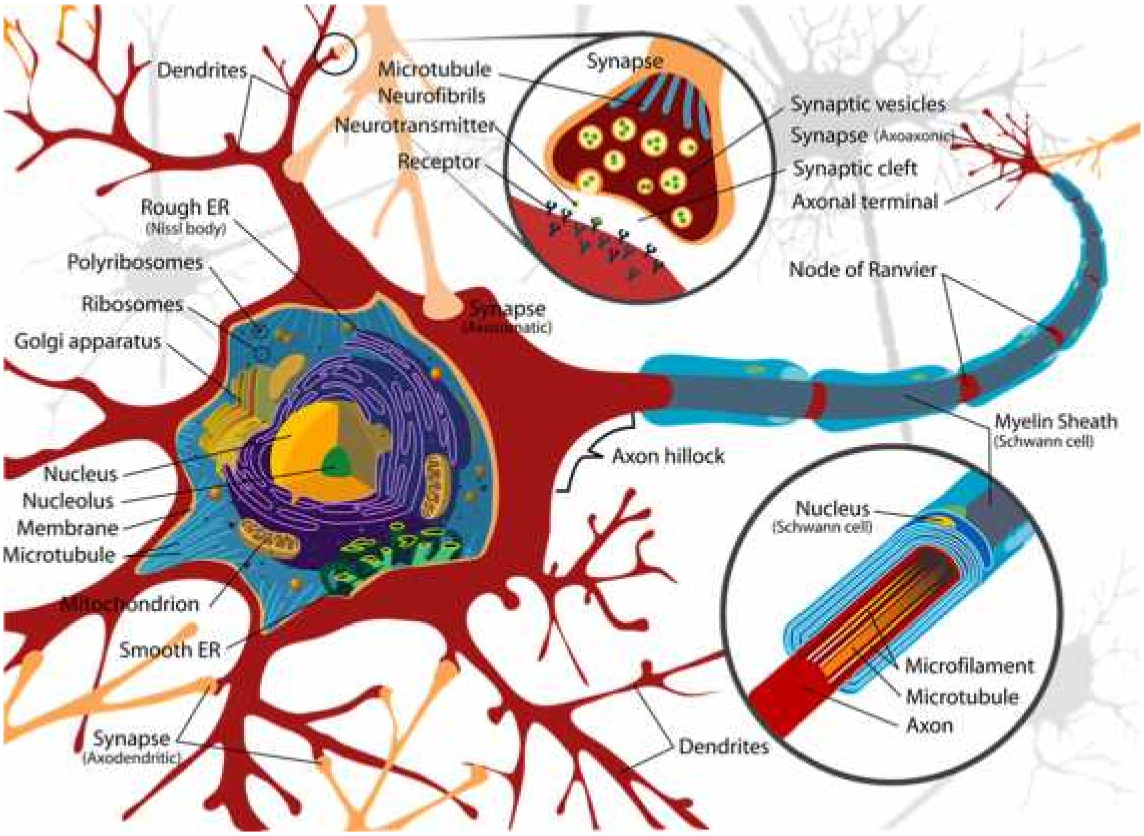


Figure 2. Representation of human nerve cell

Neuron is the fundamental unit of the neural network architecture. Dendrites, soma and axon constitute the neuron structure. Signals from surrounding neurons are received by a tree like structure called Dendrites. Each of the line is connected to one neuron. The signal from one neuron to others is transmitted by a thin cylindrical structure is called an Axon. Synapse is responsible for making contact between at the end of axon and dendrites. At the synapse, the inter neuronal signal is propagated through chemical diffusion or by electrical impulses. Only if certain condition is met, the neuron fires an electrical impulse [2]. If the total weight of the synapses which receive impulses in the period of latent summation become more than the threshold then it causes neuron fire [3].

2. Modelling the architecture of Artificial Neural Network

The information processing unit of the basic neural network architecture is a neuron. Non-linear model of a neuron is depicted in the Figure 3.

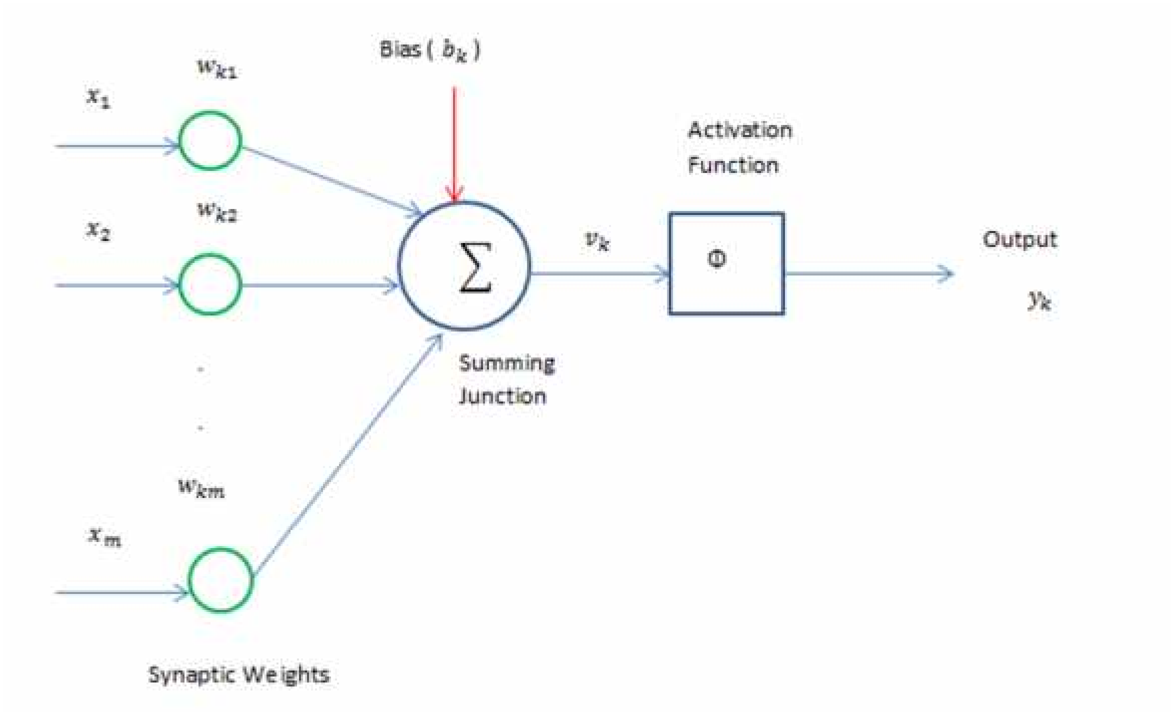


Figure 3. Representation of non-linearity of a neuron

The given neural model has three basic elements i.e. an adder, set of synapses and an activation function. The characterization of the connecting links or set of synapses are done by its own weight or strength. Signal x_j at the input of synapse j which is connected to the neuron k is further multiplied by the synaptic weight w_{kj}. The synaptic weight of the artificial neuron lies between the both positive and negative values. An adder performs the operation of the linear combination. It sums up the input signals which are weighted by the synaptic strengths. The main application of the activation function is to limit the amplitude of the output of neuron. The bias which is shown in the Figure 3 is applied externally and its main function is to decrease or increase the net input of the activation function.

In the form of mathematical expression, we can define the neuron k shown in the Figure 3 with the help of the pair of equations:

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \qquad (1)$$

and, $$y_k = \varphi \left(u_k + b_k \right) \qquad (2)$$

3. Machine Learning and its classifications

Machine learning technique give priorities to the development and execution of the computer programs that can process data and use that particular data to learn for themselves. It's just like before solving calculus exercise numerical problems, you go through the various examples given in that particular chapter i.e. you are training your mind and when you solve new question it is called testing your mind and the result you get for that particular question measures the accuracy of your performance. So we can define Machine learning as a subset of artificial intelligence (AI) which provides the given systems or state the ability to learn by itself and further modify from the experience without being explicitly programmed [4-8]. Machine learning classified into three main types i.e. reinforcement learning, supervised learning and unsupervised learning which will be discussed one by one in the following sub sections.

3.1 Supervised Learning

Supervised learning can be referred to as the situation of learning a concept with a teacher. The working of supervised machine learning is shown in the Figure 4. We may assume that the teacher possess the knowledge of the given environment which is unknown to the neural network architecture. The representation of the knowledge is done by a set of input-output examples.

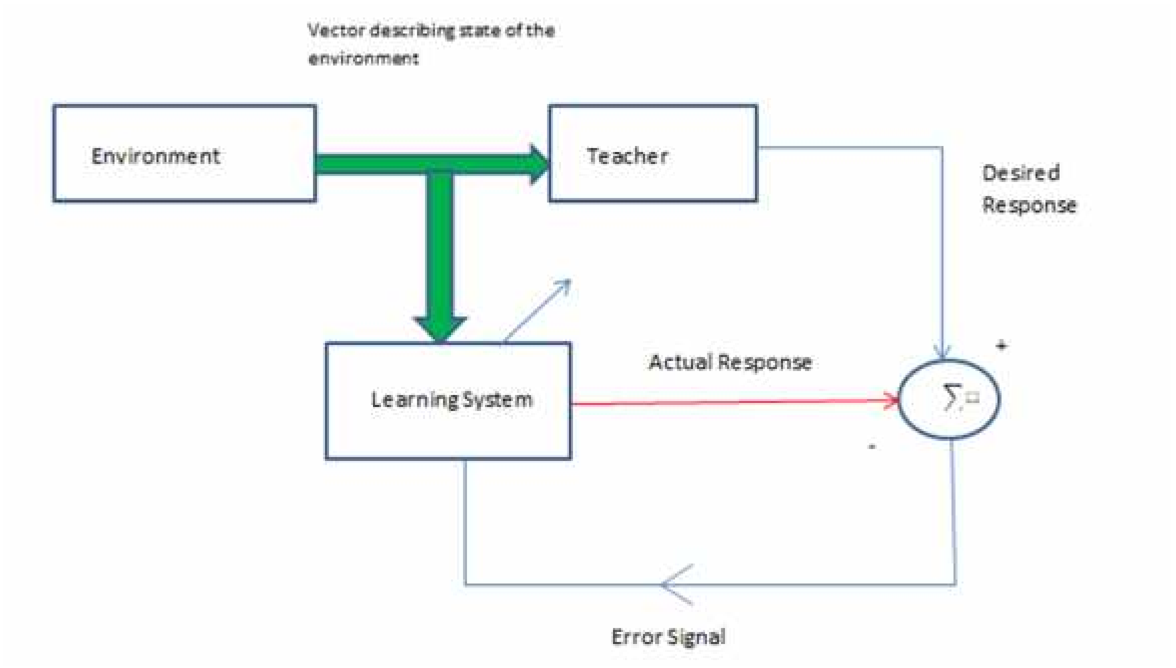


Figure 4. Working of Supervised Machine Learning

Figure 4 shows that a training vector which is dragged from the same environment is exposed to neural network and the teacher. So, for that training vector, a desired response is given by the teacher with the assistance of the possessed knowledge. Error signal shown in the Figure 4 is the difference of the actual response and the desired response. Under the combined influence of the error signal and the training vector, network parameters are adjusted. This step by step adjustment is carried out further which makes proper transfer of information of the environment possessed by the teacher to the neural network.

3.2 Unsupervised Learning

Unsupervised machine learning is a self-organized learning process in which there is no involvement of external person or teacher to look after or observe the training process. The working of the unsupervised machine learning process is shown in the Figure 5. In order to carry out unsupervised machine learning task, a competitive learning rule is used. Just take an example of a neural network consisting of two layers i.e. an input layer and a competitive layer. The available data is received by the neural network. If we talk about the competitive layer, it consists of various neurons which compete against each other for the opportunity for responding the features contained in the input data as per the given learning rule.

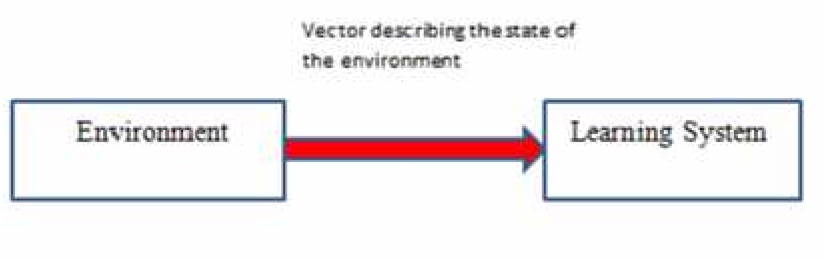


Figure 5. Working principle of Unsupervised Machine Learning

3.3 Reinforcement Learning

Reinforcement learning is a semi-supervised machine learning method which trains the models to make a sequence of decisions. The working of the Reinforcement learning is shown in the Figure 6. In an uncertain, potentially complex environment agent learns to achieve a particular goal. An artificial intelligence is subjected to a game-like situation where the computer employs trial and error method to come up with a solution to the given problem. The main objective of the Reinforcement Learning is the maximization of the total reward. Reinforcement Learning works on the rewards and penalties scheme. The Reinforcement Learning model tries to figure out the method to perform the given task in order to maximize the reward, starting from totally random trials and finishing with advanced tactics and superhuman skills.

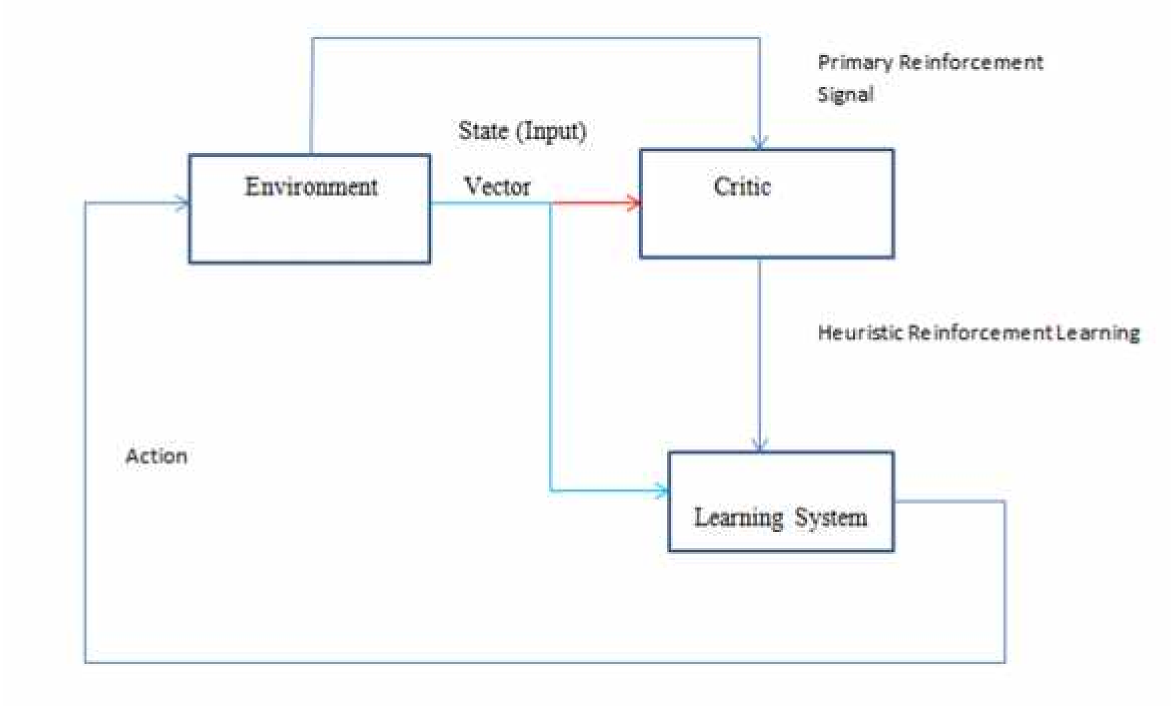
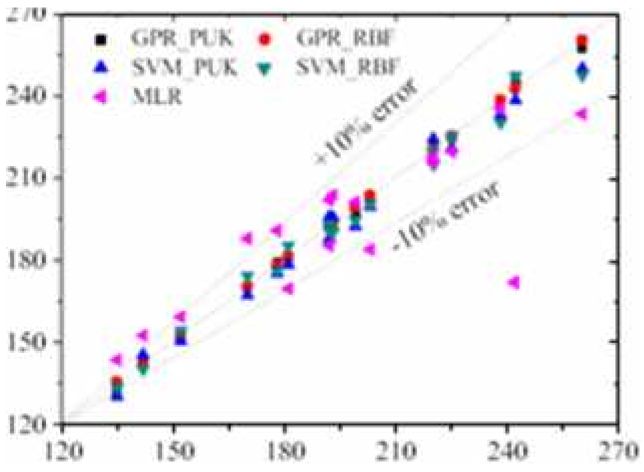

Primary Reinforcement Signal
State (Input)
Environment
Vector
Critic
Heuristic Reinforcement Learning
Action
Learning System

Figure 6. Working principle of Reinforcement Learning

4. Application of Machine Learning in Friction Stir Welding Process

Verma et al. [9] used various sophisticated machine learning approaches like . Support Vector Machining (SVM) , Gaussian Process (GP) regression, and multi-linear regression (MLR) for evaluating the friction stir welding process.

Predicted UTS (MPa)



Actual UTS (MPa)

Figure 7. Predicted vs actual values of Ultimate Tensile Strength (UTS) using SVM, GP, and MLR using training data [9].

As shown in Figure 7 it is observed that in comparison to the SVM and MLR models, the GPR approach works better. Therefore, GPR approach is generally used successfully for prediction of the UTS of Friction Stir welded joints.

Debroy et al. [10] studied the conditions for void formation using a Bayesian Neural Network and a decision tree. Schematic representation of the research is shown in the Figure 8. The study showed that the neural network and the decision tree predicted void formation with 96.6% accuracy by computing the causative variables like temperature, strain rate, torque, and maximum shear stress on the tool pin.

Celik et al. [11] investigated the correlation between the friction welding parameters and tensile strength of both AISI 316 austenitic-stainless steel and Ck 45 steel by developing an Artificial Neural Network (ANN) model. Figure 9 represents the artificial neural network architecture used in the study.

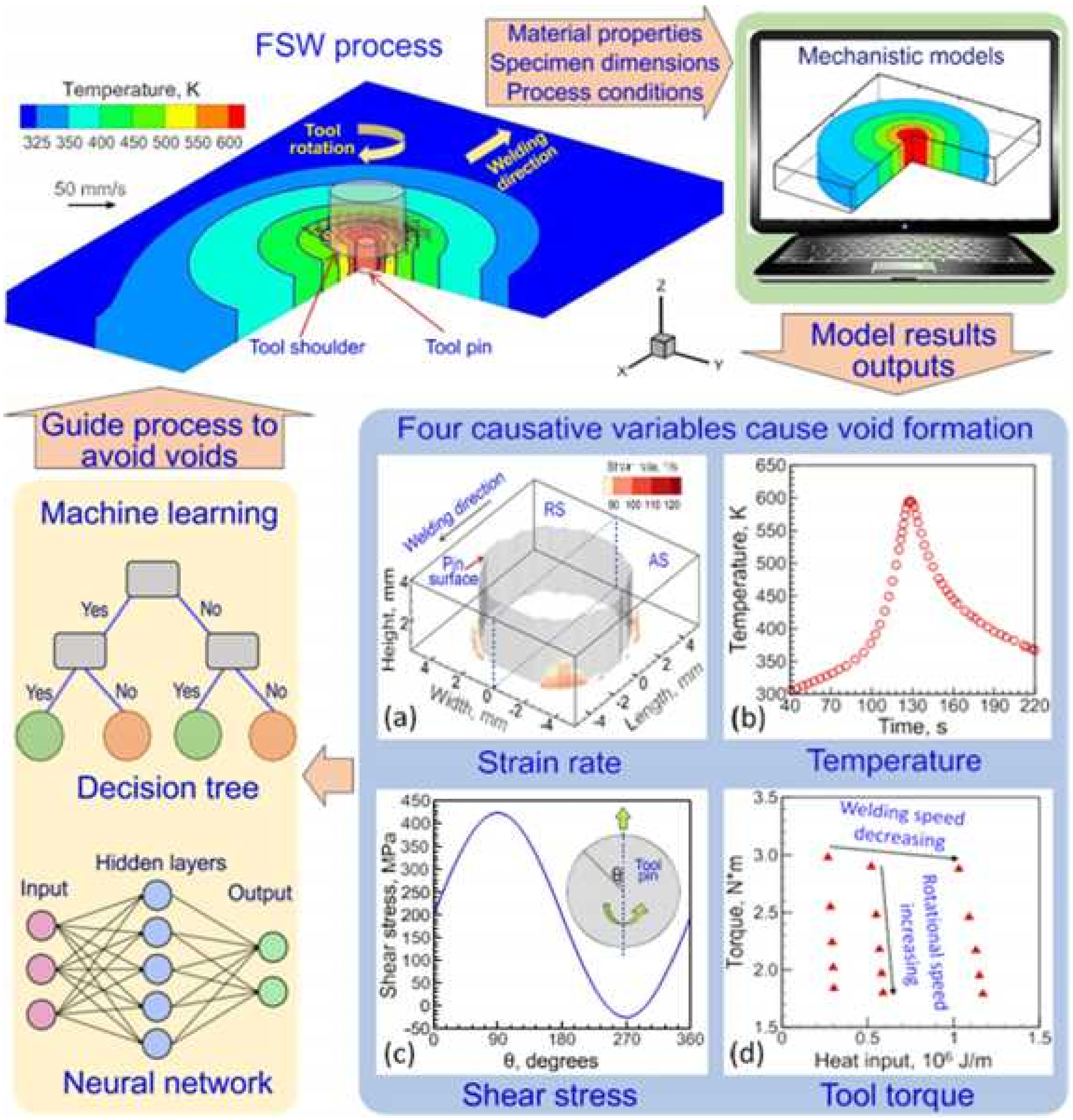


Figure 8. The components are FSW process, mechanistic models, and machine learning methods (neural network and decision tree) [10].

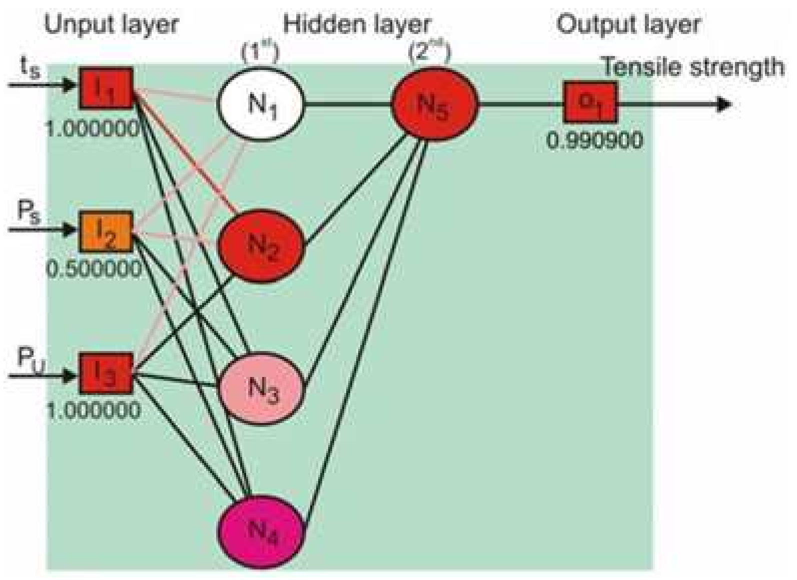


Figure 9. Four layered neural network architecture [11]

Figure 10 shows that a good correlation was obtained between the Artificial Neural Network predicted values and the experimental values.

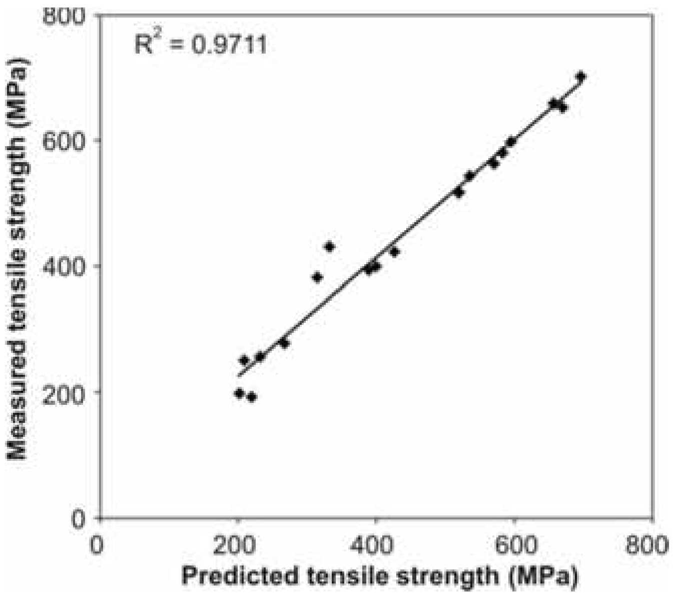


Figure 10. Comparison of measured and predicted outputs for tensile strength [11]

Maleki et al. [12] developed Artificial Neural Network based on backward propagation algorithm for predicting the yield strength, tensile strength, notch-tensile strength and hardness of friction stir welded AA 7075-T6 joints. The Neural Network architecture used in the study is shown in the Figure 11.

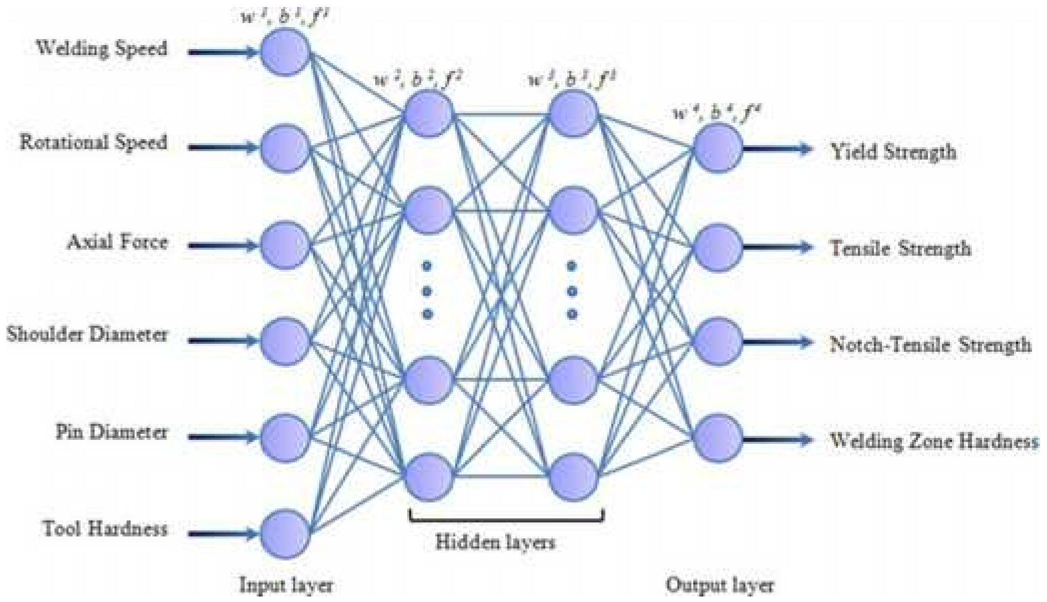


Figure 11.Schematic representation of Neural Network architecture used by Maleki et al. [12]

Figure 12 shows the comparison of the predicted and the experimental results. The results show that the if neural networks are adjusted carefully then it can be used for modeling of various Friction Stir Welding parameters.

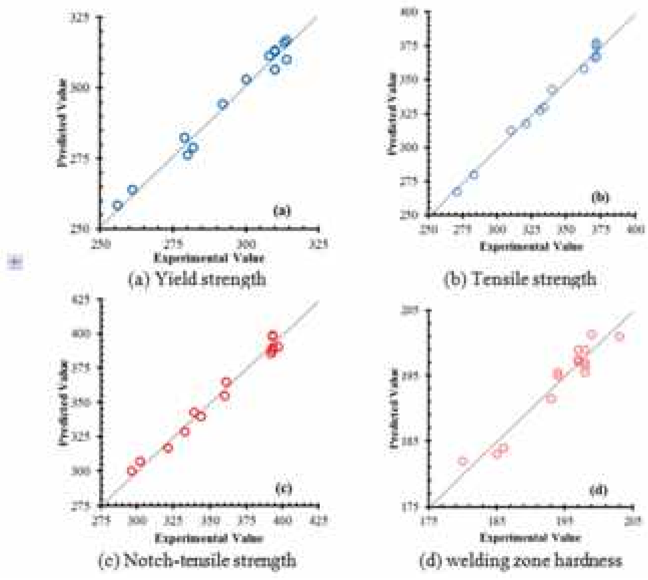


Figure 12. Experimental values plotted against predicted values [12]

Mishra et al. [13] implemented the Convolutional Neural Network for identification of the texture of Friction Stir Welded joints and Conventional Welded joints. Macias et al. [14] established a correlation between the acoustic emission signals and the various main parameters of friction stir welding process based on artificial neural networks (ANNs) trained on Levenberg-Marquardt algorithm. Figure 13 and 14 shows the methodology and the development of Artificial Neural Network architecture respectively.

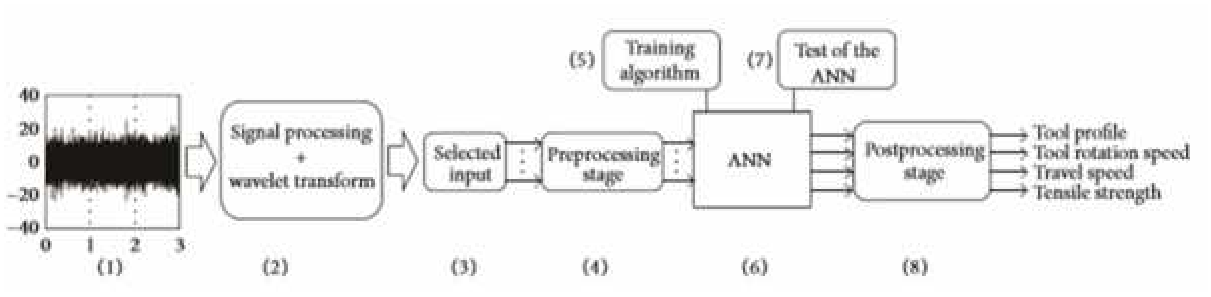


Figure 13. Artificial Neural Network development methodology [14]

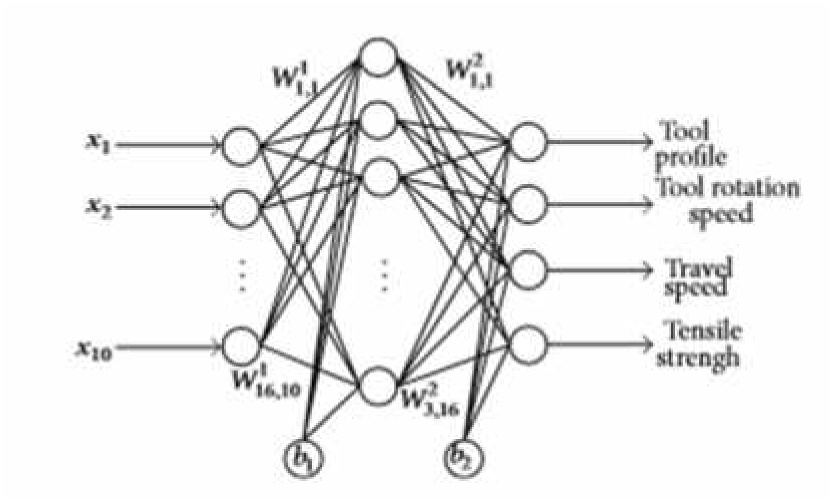


Figure 14. Artificial Neural Network architecture [14]

Figure 15 shows the predicted and measured results on the given dataset. It is clearly observed that the results obtained from the new model obtained, based on Neural Network architecture is an effective technique for the prediction of Friction Stir Welding process parameters and tensile strength of joint.

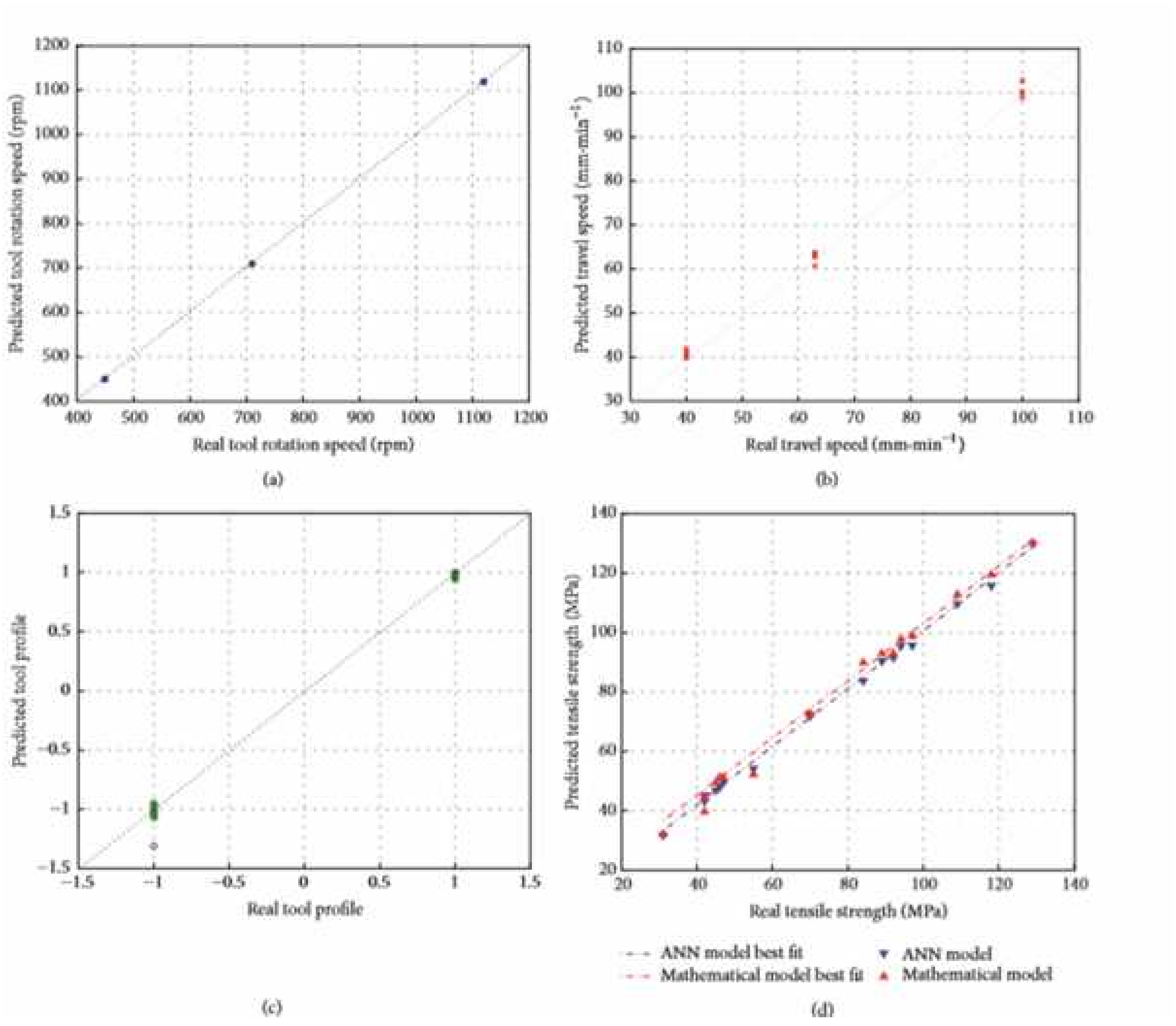


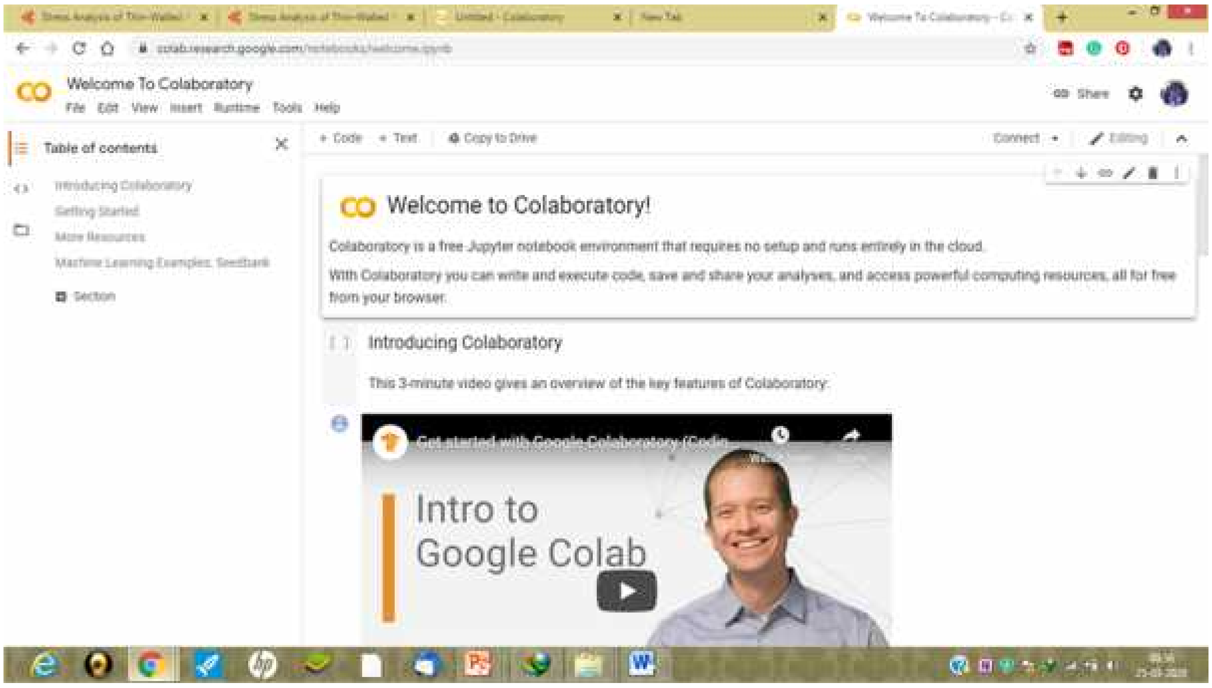
Figure 15. Predicted and Measured Values of the experimental dataset [14].

5. Designing a simple Artificial Neural Network on Google Colab by using Python coding

In this section, development of an Artificial Neural Network model by using Python Programming on Google Colaboratory platform as shown in Figure 16 will be discussed. Table 1 shows the experimental datset which shows the Ultimate Tensile Strength (MPa) value of Friction Stir Welded joints against the tool rotational speed of the tool (rpm).

Table 1. Experimental Dataset

Tool Rotational Speed (RPM)	UTS (MPa)
425	465
575	444
350	440.6
650	415
500	448


colab.research.google.com/notebooks/welcome.ipynb
Welcome To Colaboratory
File Edit View Insert Runtime Tools Help
Share
Table of contents
Introducing Colaboratory
Getting Started
More Resources
Machine Learning Examples: Seedbank
Section
+ Code + Text Copy to Drive
Connect
Editing
Get started with Google Colaboratory (Codin...
Intro to Google Colab

Welcome to Colaboratory!

Colaboratory is a free Jupyter notebook environment that requires no setup and runs entirely in the cloud.

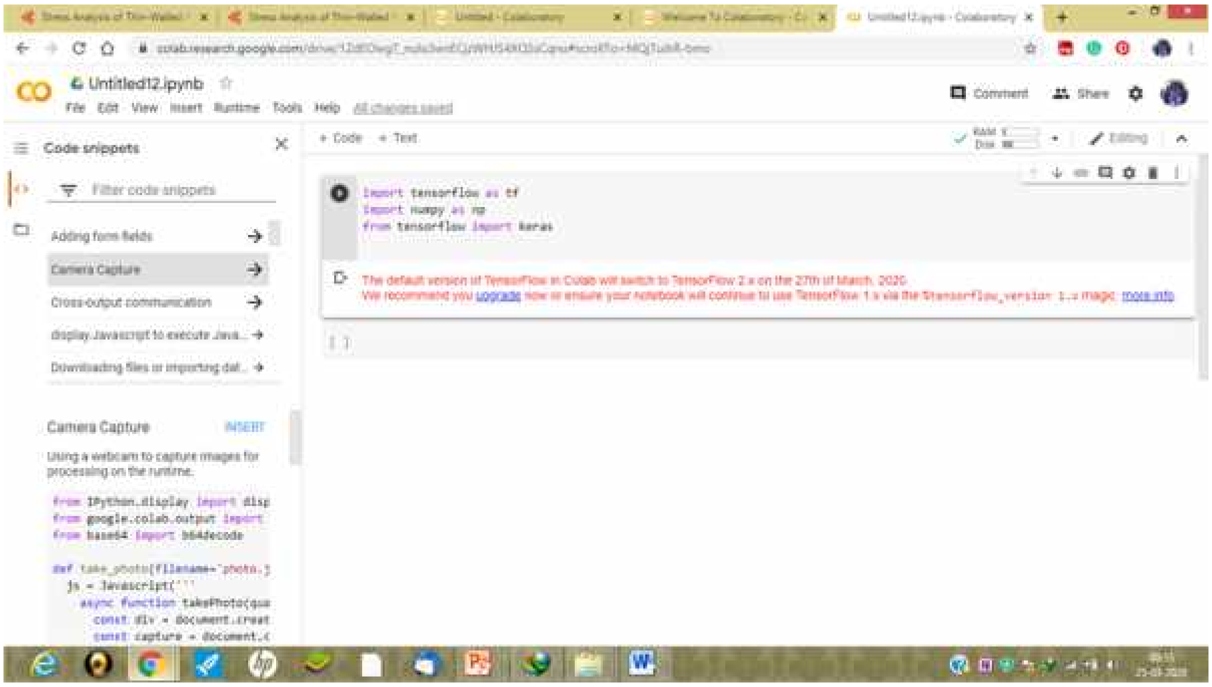
With Colaboratory you can write and execute code, save and share your analyses, and access powerful computing resources, all for free from your browser.

Introducing Colaboratory

This 3-minute video gives an overview of the key features of Colaboratory:

Figure 16. Google Colaboratory Platform

First we will start with our imports. Here we are importing TensorFlow and calling it tf for ease of use. We then import a library called numpy, which helps us to represent our data as lists easily and quickly. The framework for defining a neural network as a set of Sequential layers is called keras, so we import that too as shown in the Figure 17.

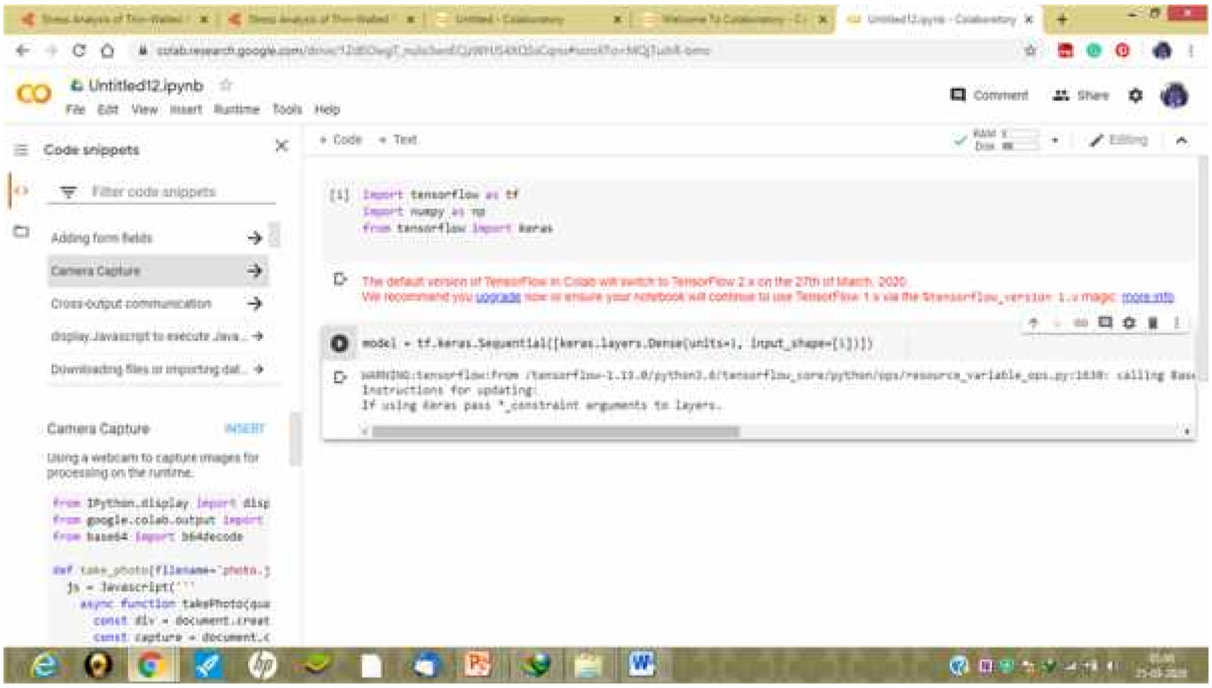


```
import tensorflow as tf
import numpy as np
from tensorflow import keras
```

```
The default version of TensorFlow in Colab will switch to TensorFlow 2.x on the 27th of March, 2020.
We recommend you upgrade now or ensure your notebook will continue to use TensorFlow 1.x via the %tensorflow_version 1.x magic: more info
```

Figure 17. Importing process

Next we will create the simplest possible neural network. It has 1 layer, and that layer has 1 neuron, and the input shape to it is just 1 value as shown in the Figure 18.



```
[1] import tensorflow as tf
    import numpy as np
    from tensorflow import keras
```

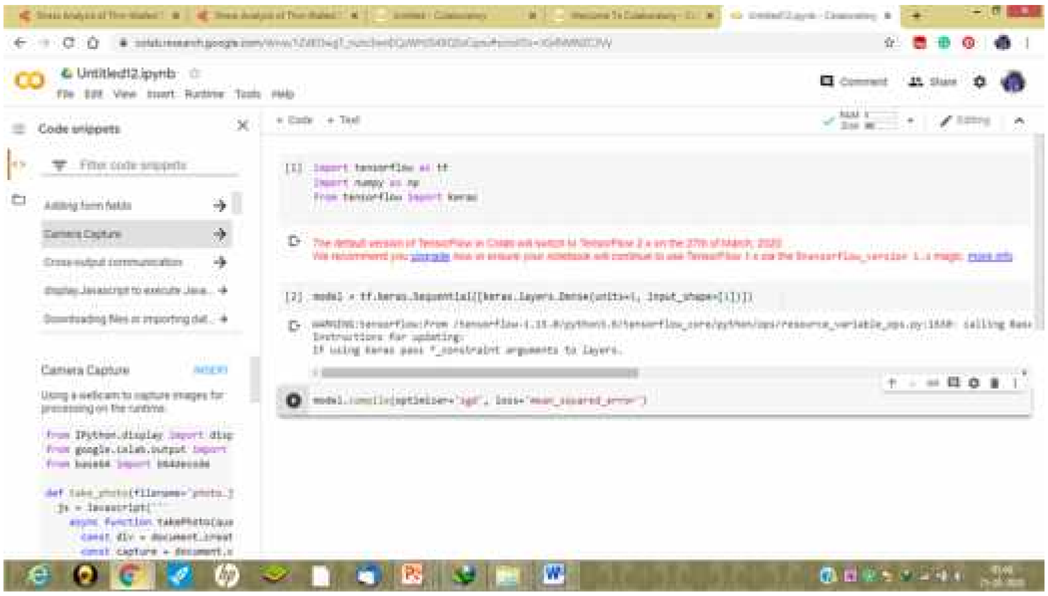
```
The default version of TensorFlow in Colab will switch to TensorFlow 2.x on the 27th of March, 2020.
We recommend you upgrade now or ensure your notebook will continue to use TensorFlow 1.x via the %tensorflow_version 1.x magic: more info
```

```
model = tf.keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])
```

```
WARNING:tensorflow:From /tensorflow-1.15.0/python3.6/tensorflow_core/python/ops/resource_variable_ops.py:1630: calling Bas
Instructions for updating:
If using Keras pass *_constraint arguments to layers.
```

Figure 18. Defining the Neural Network

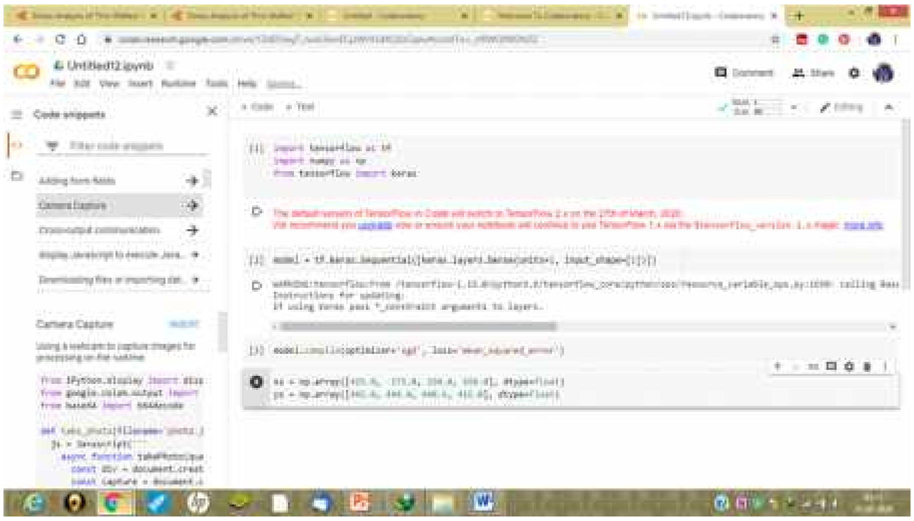
Now we compile our Neural Network as shown in Figure 19. When we do so, we have to specify 2 functions, a loss and an optimizer. The LOSS function measures the guessed answers against the known correct answers and measures how well or how badly it did. It then uses the OPTIMIZER function to make another guess. Based on how the loss function went, it will try to minimize the loss. It will repeat this for the number of EPOCHS which you will see shortly. But first, here's how we tell it to use 'MEAN SQUARED ERROR' for the loss and 'STOCHASTIC GRADIENT DESCENT' for the optimizer.



```
[1] import tensorflow as tf
    import numpy as np
    from tensorflow import keras

The default version of TensorFlow in Colab will switch to TensorFlow 2.x on the 27th of March, 2020.
We recommend you upgrade now or ensure your notebook will continue to use TensorFlow 1.x via the %tensorflow_version 1.x magic: more info

[2] model = tf.keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])

WARNING:tensorflow:From /tensorflow-1.15.0/python3.6/tensorflow_core/python/ops/resource_variable_ops.py:1630: calling Bas
Instructions for updating:
If using Keras pass *_constraint arguments to layers.

model.compile(optimizer='sgd', loss='mean_squared_error')
```

Figure 19. Compiling the Neural Network

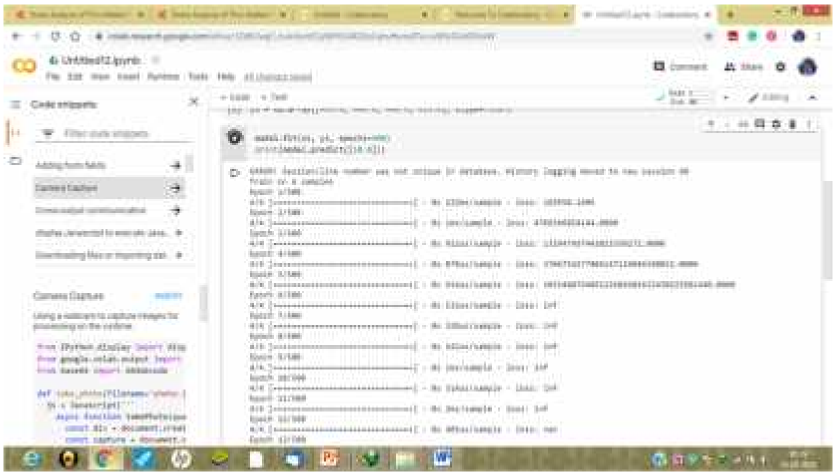
Next up we'll feed in some data as shown in the Figure 20. A python library called 'Numpy' provides lots of array type data structures that are a defacto standard way of doing it. We declare that we want to use these by specifying the values as an np.array[].



```
[1] import tensorflow as tf
    import numpy as np
    from tensorflow import keras

The default version of TensorFlow in Colab will switch to TensorFlow 2.x on the 27th of March, 2020.
We recommend you upgrade now or ensure your notebook will continue to use TensorFlow 1.x via the %tensorflow_version 1.x magic: more info

[2] model = tf.keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])

WARNING:tensorflow:From /tensorflow-1.15.0/python3.6/tensorflow_core/python/ops/resource_variable_ops.py:1630: calling Bas
Instructions for updating:
If using Keras pass *_constraint arguments to layers.

[3] model.compile(optimizer='sgd', loss='mean_squared_error')

xs = np.array([[illegible]], dtype=float)
ys = np.array([[illegible]], dtype=float)
```

Figure 20.Providing the data

The process of training the neural network, where it 'learns' the relationship between the Xs and Ys is in the model.fit call which is shown in Figure 21. This is where it will go through the loop we spoke about above, making a guess, measuring how good or bad it is (aka the loss), using the opimizer to make another guess etc. It will do it for the number of epochs you specify. When you run this code, you'll see the loss on the right hand side.



```
model.fit(xs, ys, epochs=500)
print(model.predict([10.0]))
```

```
[illegible]
Train on 6 samples
Epoch 1/500
6/6 [==============================] - 0s [illegible]/sample - loss: [illegible]
Epoch 2/500
6/6 [==============================] - 0s [illegible]/sample - loss: [illegible]
Epoch 3/500
6/6 [==============================] - 0s [illegible]/sample - loss: [illegible]
Epoch 4/500
6/6 [==============================] - 0s [illegible]/sample - loss: [illegible]
Epoch 5/500
6/6 [==============================] - 0s [illegible]/sample - loss: [illegible]
Epoch 6/500
6/6 [==============================] - 0s [illegible]/sample - loss: inf
Epoch 7/500
6/6 [==============================] - 0s [illegible]/sample - loss: inf
Epoch 8/500
6/6 [==============================] - 0s [illegible]/sample - loss: inf
Epoch 9/500
6/6 [==============================] - 0s [illegible]/sample - loss: inf
Epoch 10/500
6/6 [==============================] - 0s [illegible]/sample - loss: inf
Epoch 11/500
6/6 [==============================] - 0s [illegible]/sample - loss: inf
Epoch 12/500
6/6 [==============================] - 0s [illegible]/sample - loss: nan
Epoch 13/500
```

Figure 21. Training and Testing the Neural Network

6. Conclusion

There is a loss of time and materials if the optimization of the Friction Stir Welding parameters is done through experimental studies which further leads to increase in the cost of the experiment. Machine Learning approach like Artificial Neural Network and image processing overcome these issues. So, it can be concluded that the mechanical and microstructure properties can be predicted and also the defects formation can also be observed by the implementation of various Machine Learning tools in the Friction Stir Welding process.

References

[1]. Haykin, S.S., 2009. *Neural networks and learning machines*/Simon Haykin.

[2]. Zurada, Jacek M.``Introduction to Artificial Neural System". *West Publishing Company*, St. Paul, MN, 1992

[3]. Arbib, Michael A. ``Brains, Machines, and Mathematics: Second Edition". *Springer-Verlag*, New York, NY, 1987.

[4]. Alpaydin, E., 2020. *Introduction to machine learning*. MIT press.

[5]. Mitchell, T.M., 1997. *Machine learning*.

[6]. Marsland, S., 2015. *Machine learning: an algorithmic perspective*. CRC press.

[7]. Michie, D., Spiegelhalter, D.J. and Taylor, C.C., 1994. *Machine learning. Neural and Statistical Classification,* 13(1994), pp.1-298.

[8]. Quinlan, J.R., 2014. C4. 5: *programs for machine learning*. Elsevier.

[9]. Verma, S., Gupta, M. and Misra, J.P., 2018. Performance evaluation of friction stir welding using machine learning approaches. *MethodsX*, 5, pp.1048-1058.

[10]. Du, Y., Mukherjee, T. and DebRoy, T., 2019. Conditions for void formation in friction stir welding from machine learning. npj *Computational Materials*, 5(1), pp.1-8.

[11]. Ersozlu, I. and Celik, S., 2019. Artificial Neural Network application to the friction welding of AISI 316 and Ck 45 steels. *Kovove Materialy-Metallic Materials*, 57(3), pp.199-205.

[12]. Maleki, E., 2015. Artificial neural networks application for modeling of friction stir welding effects on mechanical properties of 7075-T6 aluminum alloy. *In IOP Conference Series: Materials Science and Engineering* (Vol. 103, No. 1, p. 012034). IOP Publishing.

[13]. Mishra, A. and Nagpal, K., 2019. Convolutional Neural Network for Image Processing of Friction Stir Welded and Conventional Welded Joints Texture. *Int. J. Hum. Comp. Inter. Data Min,* 2(1&2), pp.5-9.

[14]. Jiménez-Macías, E., Sánchez-Roca, A., Carvajal-Fals, H., Blanco-Fernández, J. and Martínez-Cámara, E., 2014. Wavelets application in prediction of friction stir welding parameters of alloy joints from vibroacoustic ANN-based model. *In Abstract and Applied Analysis* (Vol. 2014). Hindawi.

Author'sBiography

Akshansh Mishra received his graduation degree in Mechanical Engineering from SRM Institute of Science and Technology in 2017 & he is now admitted to MS in Mechanical Engineering degree in Politecnico Di Milano (QS world rank 9 in Mechanical Engineering). He also founded a research and development firm known as Stir Research Technologies which works on the collaborative research in Artificial Intelligence and Friction Stir Welding. His main research interests are Machine Learning, Reinforcement Learning, Advanced Manufacturing Process, and Friction Stir Welding.

How to Cite

Mishra. Akshansh(2019). Understanding Machine Learning for Friction Stir Welding Technology, *International Journal of Machine Learning and Networked Collaborative Engineering,* Vol. 03 No. 3, 2019, pp.143-158.

doi : https://doi.org/10.30991/IJMLNCE.2019v03i03.003