

Deep Learning Models, Open Source Tools: A Review

Seema R.Wankhade

*Department of Information Technology
Government College of Engineering, Amravati, Maharashtra, India*

Anjali B.Raut

*Department of Computer Science and Engineering
HVPM College of Engineering and Technology, Amravati, Maharashtra, India*

Abstract - In the last few years, machine learning and data mining have become popular topics among the research community. Deep learning is an emerging area of machine learning (ML) research. In deep learning, input is given, while processing iteration continues until output reached an acceptable level of accuracy. Deep Learning can create a more accurate model from the large quality of unlabeled and unstructured data (Big Data). The advantage of deep learning concerning supervised learning is it's faster and more accurate. In this paper, we present a comprehensive review of deep learning models and open source tools used in deep learning applications.

Keywords: Machine Learning, Deep learning, Information Processing, Big Data

I. INTRODUCTION

Now a day's Deep learning is getting more and more attention, defiantly for good reason. It's achieving unprecedented levels of accuracy. Deep learning is a subset of machine learning where a model learns to perform classification from images, text, or sound. Deep learning is usually implemented using neural network architecture. Traditional neural networks contain only 2 or 3 layers; while deep networks can have multiple layers maybe hundreds of layers. Due to which advanced tools and techniques have dramatically improved deep learning algorithms.

1.1 An overview of Deep Learning

Deep learning includes statistics and predictive modeling. In traditional machine learning, the learning process is supervised, and the programmer has to be extremely specific when telling the computer what types of things it should be looking for to decide if an image contains a particular object or does not contain an object. This is a tedious process called feature extraction, and the accuracy depends entirely upon the programmer's ability to accurately define a feature set for an object. The benefits of deep learning are the program builds the feature set by itself without any supervision. Unsupervised learning is faster, as well most of the time it's more accurate. Why Deep? It refers to the total number of layers (more layers).

1.2 Deep Learning Terminology and Definitions

Deep neural networks are inspired by biological nervous systems. It consists of an input layer, a number of hidden layers, and an output layer. The layers are interconnected with each other through nodes, called neurons, where every hidden layer uses the output of the previously hidden layer as an input.

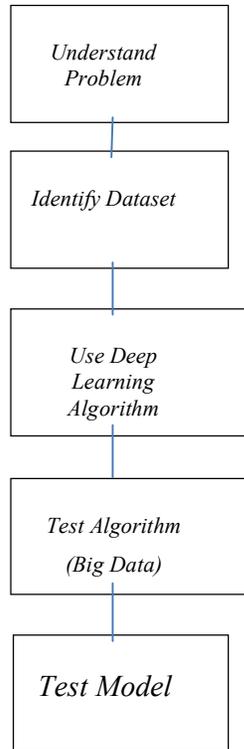


Fig. 1.1 Basic Deep Learning Model Approach

Deep learning is a process that employs deep neural network architectures, which are particular types of machine learning algorithms. Deep neural networks are neural networks with one hidden layer minimum or more than one hidden layer. Deep learning refers to a process, which employs deep neural network architectures, which are specific types of machine learning algorithms.

II. DEEP LEARNING MODELS

In Deep learning, there are algorithms that train machines by learning from examples. In a neural network, a lot of artificial neurons (nodes) are there which is structured like the human. These nodes are placed next to each other in the following three layers:

- 1) The input layer
- 2) The hidden layer or Layers
- 3) The output layer

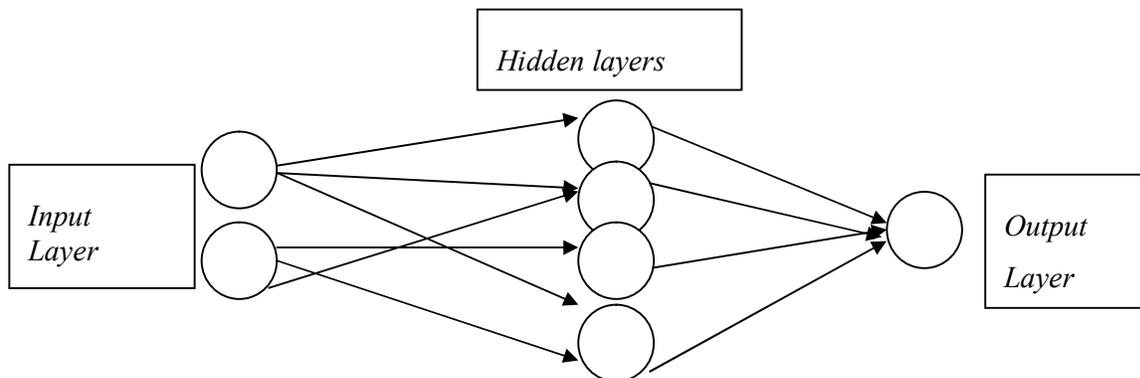


Fig. 2.1 Simple Neural Network

2.1 Convolutional Neural Networks (CNNs)

CNN's have multiple layers that process and extract features from raw data. CNN features a convolution layer that has several filters to perform the convolution operation. CNN's have a ReLU layer to perform operations on elements. The output may be a rectified feature map[1].

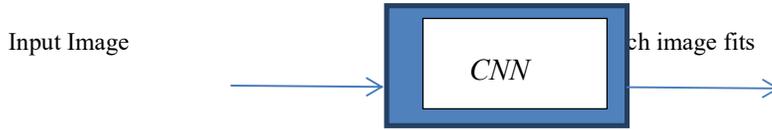


Fig:2.2 Convolutional Neural Networks

2.1.1. Pooling Layer: The rectified feature map next feeds into a pooling layer. Pooling may be a down-sampling operation that reduces the size of the feature map. The pooling layer then converts the resulting two-dimensional arrays from the pooled feature map into one , long, continuous, linear vector by flattening it.

2.1.2 Fully Connected Layer: A fully connected layer forms when the flattened matrix from the pooling layer is fed as an input, which classifies and identifies the images.

2.2 Long Short Term Memory Networks (LSTMs)

LSTMs are type of Recurrent Neural Network (RNN) which can learn and memorize long-term dependencies. Recalling previous stage data for long periods is actually the default behavior. LSTMs remember information over time. LSTM are useful in time-series prediction as they remember previous inputs. LSTMs are specifically used for speech recognition, music composition, and pharmaceutical development[2].

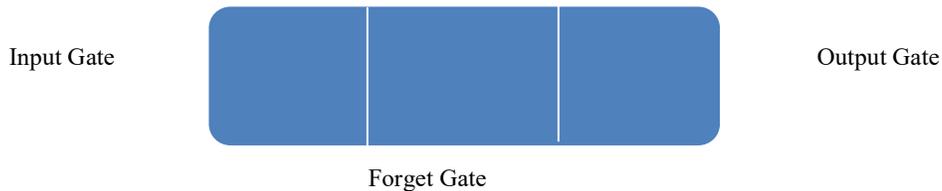


Fig2.3: Long Short Term Memory Networks

Forget parts not in use of the previous state. Update the cell-state with newer values. The outputs from the LSTM to be fed as inputs to the current phase. As the output from the LSTM becomes an input to the ongoing phase and can memorize previous inputs due to its internal memory unique carecteristic.

2.3 Recurrent Neural Networks (RNNs)

RNNs are commonly used for image captioning, time-series analysis, natural-language processing, handwriting recognition, and machine translation. In RNN the output at time t-1 feeds into the input at time t. Similarly, the output at time t feeds into the input at time t+1. RNNs can process inputs of any size. And also the model size does not increase with the input size[3].

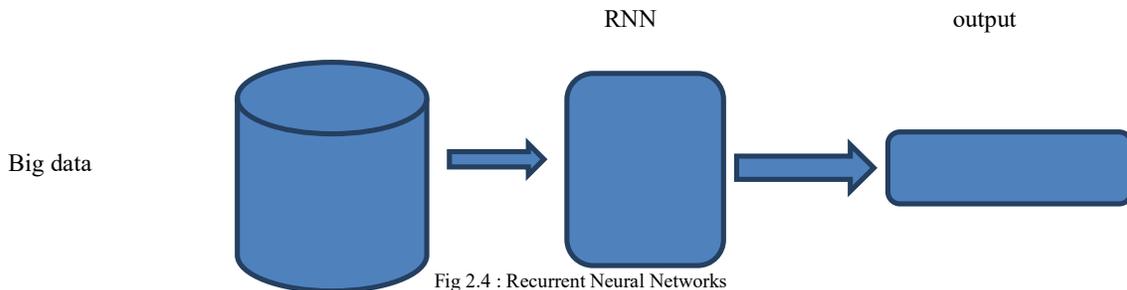


Fig 2.4 : Recurrent Neural Networks

2.4 Generative Adversarial Networks (GANs)

GAN comprises two components: a generator learns to generate fake data, a discriminator, learns from that false information. GANs are used to improve astronomical images and simulate gravitational lensing for dark-matter research. For developing video games GANs are used to increase the resolution to provide real effect, GANs generate realistic images and cartoon characters[4].

GANs work as below: The discriminator learns to distinguish between the generator's fake data and the real sample data. During training, the generator produces fake data, and the discriminator quickly identifies and learns it's false. The GAN sends the output to the generator and the discriminator to update the model.

Below is a diagram of how GANs operate:

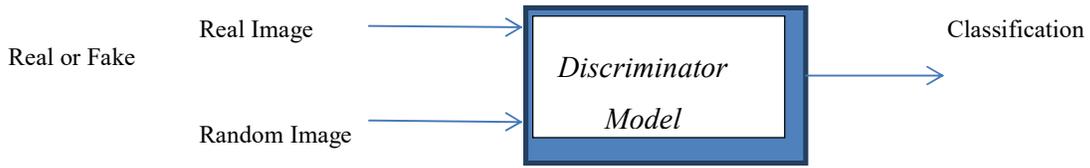


Fig 2.5 Generative Adversarial Networks

2.5 Radial Basis Function Networks (RBFNs)

RBFNs are special sorts of feed forward neural networks which uses radial basis functions as activation functions. RBFNs have an input vector that feeds to the input layer. They have a layer of RBF neurons. The function finds the weighted sum of the inputs, and therefore the output layer has one node per category or class of knowledge. The neurons within the hidden layer contain the Gaussian transfer functions, which have outputs that are inversely proportional to the space from the neuron's center. The network's output is a linear combination of the input's radial-basis functions and the neuron's parameters[5].

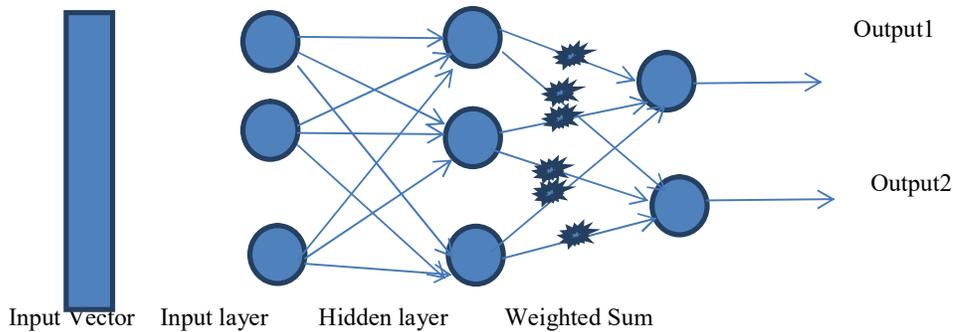


Fig 2.6 Radial Basis Function Networks

2.6. Multilayer Perceptron's

These are feed forward neural networks with multiple layers of perceptrons with activation functions. MLPs contains an input layer and an output layer fully connected with one another. The amount of input and output layers is that the same but may have multiple hidden layers.

MLPs feed the info to the input layer of the network. MLPs compute the input with the weights that exist between the input layer and therefore the hidden layers. MLPs use activation functions to work out which

nodes to fire or activate. MLPs train the model to know the correlation and learn the dependencies between the independent and therefore the target variables from a training data set[6].

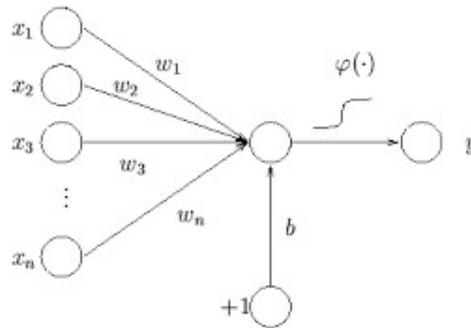


Fig. 2.7 Multilayer Perceptrons

2.7. Self Organizing Maps (SOMs)

Data visualization attempts to unravel the matter that humans cannot easily visualize i.e. high-dimensional data. SOMs are designed to help users to understand this high-dimensional information. SOMs initialize weights for every node and choose a vector randomly from the training data. SOMs examine every node to find which weights is the most likely input vector. The closer node is named the Best Matching Unit (BMU).

2.8 Deep Belief Networks (DBNs)

DBNs are generative models with multiple layers of stochastic, latent variables. The latent variables have binary values either 1 or 0 and are often called hidden units. DBNs are a stack of Boltzmann Machines with connections between the layers, and each RBM layer communicates with both the previous and next layers. Deep Belief Networks (DBNs) are used for image-recognition, video-recognition, and motion-capture data. Greedy learning algorithms which uses layer-by-layer approach for learning the top-down, generative weights train DBNs. [8].

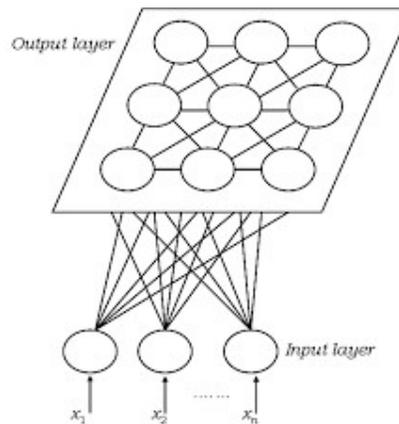


Fig.2.8: Deep Belief Networks architecture

2.9. Restricted Boltzmann Machines (RBMs)

RBMs are stochastic neural networks which will learn from a probability distribution over a group of inputs. This deep learning algorithm is employed for dimensionality reduction, classification, regression, collaborative filtering, feature learning, and topic modeling. RBMs constitute the building blocks of DBNs[9].

RBMs contain two layers :

- Visible units
- Hidden units

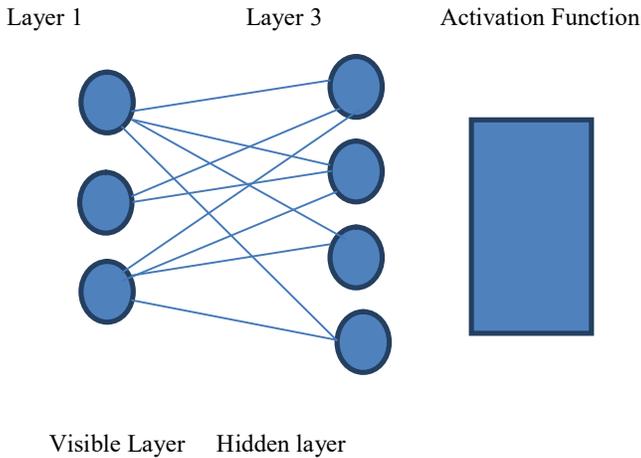


Fig 2.9 Restricted Boltzmann Machines

Each visible unit is connected to all hidden units. RBMs have a bias unit that is connected to all the visible units and the hidden units, and they have no output nodes.

2.10. Autoencoders

Autoencoders are a type of feedforward neural network where the input and output are identical. They are trained neural networks that replicate the data from the input layer to the output layer. Autoencoders are used discovery, prediction, and image processing [10].

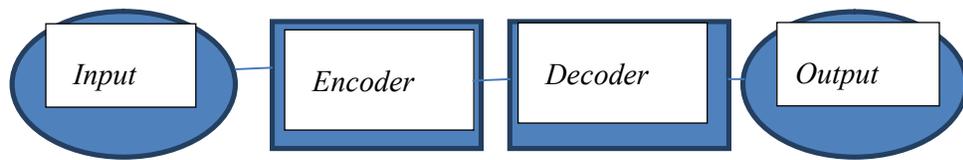


Fig 2.10 Autoencoders

III. OPEN SOURCE TOOLS USED IN MACHINE LEARNING DEEP LEARNING

Scikit-learn is for machine learning development in python. It provides a library for the Python programming language. It helps in data mining and data analysis. It provides models and algorithms for classification, regression, clustering, dimensional reduction, model selection, and pre-processing [12]. Easily understandable documentation is provided. Parameters for any specific algorithm can be changed while calling objects. It's free to use [11].

PyTorch is a Torch based, Python machine learning library. The torch is a Lua based computing framework, scripting language, and machine learning library. It helps in building neural networks through Autograd Module. It provides a variety of optimization algorithms for building neural networks. PyTorch can be used on cloud platforms. It provides distributed training [13].

TensorFlow provides a JavaScript library that helps in machine learning. APIs will help you to build and train the models. Existing models can be run with the help of TensorFlow.js which is a model converter. It helps in the neural network. It can be used in two ways, i.e. by script tags or by installing through NPM. It can even help for human pose estimation [13].

Weka these machine learning algorithms help in data mining. Weka can be used for data preparation, classification, and regression, clustering, visualization, and association rules mining. Provides online courses for training. Easy to understand algorithms [14].

KNIME is a tool for data analytics, reporting, and integration platform. Using the data pipelining concept, combines different components for machine learning and data mining. It can integrate the code of programming languages like C, C++, R, Python, Java, JavaScript, etc. It can be used for business intelligence, financial data analysis, and CRM. It can work as a SAS alternative. It is easy to deploy and install. It's Difficult to build complicated models using KNIME. It has limited visualization and exporting capabilities [15].

Google Colab is a cloud service which supports Python. It will help in building the machine learning applications using the libraries of PyTorch, Keras, TensorFlow, and OpenCV. It helps in machine learning education. Assists in machine learning research. Can be used from your googledrive [16].

Apache Mahout helps mathematicians, statisticians, and data scientists for executing their algorithms. It provides algorithms for Pre-processors, Regression, Clustering, Recommenders, and Distributed Linear Algebra. Java libraries are included for common math operations. It follows Distributed linear algebra framework. It works for large data sets. Simple . Extensible. Needs more helpful documentation. Some algorithms are missing [17].

Accord.Net Accord.Net provides machine learning libraries for image and audio processing. Provides algorithms for Numerical linear algebra. Numerical optimization Statistics, Artificial Neural networks. Image, audio, & signal processing. It also provides support for graph plotting & visualization libraries. Libraries are made available from the source code and also through executable installer & NuGet package manager. It supports only. Net supported languages [18].

Shogun Shogun provides various algorithms and data structures for machine learning. These machine learning libraries are used for research and education. It provides support vector machines for regression and classification. It helps in implementing Hidden Markov models. It offers support for many languages like – Python, Octave, R, Ruby, Java, Scala, and Lua. Pros: It can process large data-sets. Easy to use. Provides good customer support. Offers good features and functionalities [19].

Keras.io Keras is an API for neural networks. It helps in doing quick research and is written in Python. Features: It can be used for easy and fast prototyping. It supports convolution networks. It assists recurrent networks. It supports a combination of two networks. It can be run on the CPU and GPU. Pros: User-friendly Modular Extensible. Cons: To use Keras, you must need TensorFlow, Theano, or CNTK [20].

Machine Learning Tools	Platform	Written in language	Algorithms or Features
Scikit Learn	Linux, Mac OS, Windows	Python, Cython, C, C++	Classification Regression Clustering Preprocessing Model Selection Dimensionality reduction.
PyTorch	Linux, Mac OS, Windows	Python, C++, CUDA	Autograd Module Optim Module nn Module
TensorFlow	Linux, Mac OS,	Python, C++,	Provides a library for

	Windows	CUDA	dataflow programming.
Weka	Linux, Mac OS, Windows	Java	Data preparation Classification Regression Clustering Visualization Association rules mining
KNIME	Linux, Mac OS, Windows	Java	Can work with large data volume. Supports text mining & image mining through plugins
Colab	Cloud Service		Supports libraries of PyTorch, Keras, TensorFlow, and OpenCV
Apache Mahout	Cross-platform	Java Scala	Preprocessors Regression Clustering Recommenders Distributed Linear Algebra.
Accors.Net	Cross-platform	C#	Classification Regression Distribution Clustering Hypothesis Tests & Kernel Methods Image, Audio & Signal. & Vision
Shogun	Windows Linux UNIX Mac OS	C++	Regression Classification Clustering Support vector machines. Dimensionality reduction Online learning etc.
Keras.io	Cross-platform	Python	API for neural networks

Table 3.1: Open Source tools used in machine Learning/deep learning

IV. CONCLUSION

Deep learning is indeed a fast growing application of machine learning. , it is important to note that the hierarchy of layers and the supervision in learning are key factors to develop a successful application regarding deep learning. The main value of deep learning relies on the optimization of existing applications in machine learning, due to its innovativeness on the hierarchical layer processing. Deep learning is a contemporary and exciting subject of advancement in artificial intelligence.

As per the study selection of the tool depends on the requirement for the algorithm, expertise level of the user. As these libraries are free to use they are regularly updated. Tensor Flow is more popular in machine learning, but it has a learning curve. Scikit-learn and PyTorch are also popular tools for machine learning and both support Python programming language. Keras.io and Tensor Flow are good for neural networks. For researchers, this study will help in selecting tools and Architecture.

REFERENCES

- [1] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186.
- [2] F. A. Gers, J. Schmidhuber and F. Cummins, "Learning to Forget: Continual Prediction with LSTM," in *Neural Computation*, vol. 12, no. 10, pp. 2451-2471, 1 Oct. 2000, doi: 10.1162/089976600300015015.
- [3] M. Kaur and A. Mohta, "A Review of Deep Learning with Recurrent Neural Network," *2019 International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 2019, pp. 460-465, doi: 10.1109/ICSSIT46314.2019.8987837.
- [4] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta and A. A. Bharath, "Generative Adversarial Networks: An Overview," in *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53-65, Jan. 2018, doi: 10.1109/MSP.2017.2765202.
- [5] B. Walczak, D.L. Massart, "The Radial Basis Functions — Partial Least Squares approach as a flexible non-linear regression technique", *AnalyticaChimicaActa*, Volume 331, Issue 3, 1996, Pages 177-185, ISSN 0003-2670
- [6] T. Nagamine, N. Mesgarani, "Understanding the Representation and Computation of Multilayer Perceptrons: A Case Study in Speech Recognition" *Proceedings of the 34th International Conference on Machine Learning*, Sydney, Australia, PMLR 70, 2017
- [7] D. Miljković, "Brief review of self-organizing maps," *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2017, pp. 1061-1066, doi: 10.23919/MIPRO.2017.7973581.
- [8] YumingHua, JunhaiGuo and Hua Zhao, "Deep Belief Networks and deep learning," *Proceedings of 2015 International Conference on Intelligent Computing and Internet of Things*, 2015, pp. 1-4, doi: 10.1109/ICAOT.2015.7111524.
- [9] Nan Zhang, Shifei Ding, Jian Zhang, Yu Xue, "An overview on Restricted Boltzmann Machines", *Neurocomputing*, Volume 275, 2018, Pages 1186-1199, ISSN 0925-2312
- [10] LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* 521, 436–444 (2015). <https://doi.org/10.1038/nature14539>
- [11] Nelli F. (2018) *Machine Learning with scikit-learn*. In: *Python Data Analytics*. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-3913-1_8
- [12] Kramer O. (2016) *Scikit-Learn*. In: *Machine Learning for Evolution Strategies*. *Studies in Big Data*, vol 20. Springer, Cham. https://doi.org/10.1007/978-3-319-33383-0_5
- [13] Peter Goldsborough "A Tour of TensorFlow" [Xiv:1610.01178v1 \[cs.LG\]](https://arxiv.org/abs/1610.01178v1) 1 Oct 2016
- [14] Smith T.C., Frank E. (2016) *Introducing Machine Learning Concepts with WEKA*. In: Mathé E., Davis S. (eds) *Statistical Genomics. Methods in Molecular Biology*, vol 1418. Humana Press, New York, NY. https://doi.org/10.1007/978-1-4939-3578-9_17
- [15] L. Feltrin, "KNIME an Open Source Solution for Predictive Analytics in the Geosciences [Software and Data Sets]," in *IEEE Geoscience and Remote Sensing Magazine*, vol. 3, no. 4, pp. 28-38, Dec. 2015, doi: 10.1109/MGRS.2015.2496160.
- [16] T. Carneiro, R. V. Medeiros Da Nóbrega, T. Nepomuceno, G. Bian, V. H. C. De Albuquerque and P. P. R. Filho, "Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications," in *IEEE Access*, vol. 6, pp. 61677-61685, 2018, doi: 10.1109/ACCESS.2018.2874767.
- [17] P. D. and Kauser Ahmed. "A Survey on Big Data Analytics: Challenges, Open Research Issues and Tools." *International Journal of Advanced Computer Science and Applications* 7.2 (2016): n. pag. Web.
- [18] S. Mohan, P. Shanmugapriya "Survey on Big Data and Machine Intelligence Tools.pdf." *Publishing India Group* 4.2 (2016): 23–30. Print.
- [19] SONNENBURG, RATSCH, HENSCHEL, WIDMER, BEHR, ZIEN, DE BONA, GEHL, BINDER AND FRANCO, "The SHOGUN Machine Learning Toolbox" *Journal of Machine Learning Research* 11 (2010) 1799-1802
- [20] Ketkar N. (2017) *Introduction to Keras*. In: *Deep Learning with Python*. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-2766-4_7