Evaluation of Neural Network Model for Better Classification of Data and Optimum Solution of Real-World Problems

Tanjima Khanam Ema[1]
19F19012@mec.edu.om
Department of Computing, Middle East College, Al Rusayl, Muscat, Postal Code: 124, Sultanate of Oman

Furwa Asim[2]
19F19721@mec.edu.om
Department of Computing, Middle East College, Al Rusayl, Muscat, Postal Code: 124, Sultanate of Oman

Anjum Zameer Bhat[3]
azameer@mec.edu.om
Department of Computing, Middle East College, Al Rusayl, Muscat, Postal Code: 124, Sultanate of Oman

Abstract

The implementation of Artificial Intelligence in providing the optimum solution to real-world problems is exponentially growing. The ability of Artificial Intelligence that finds its implementation in a broad spectrum of areas and fields has certainly increased its demand in the recent past. The concept of Neural Networks which originally has its roots in Artificial Intelligence is also gaining significant popularity and is gradually being implemented in many fields to yield benefits, most of the time beyond expectations. Neural Networks mimic the operations of a human brain to derive various relations between the data sets. Neural Networks find their use in a wide variety of fields that includes and is not limited to finance, algorithm trading, credit risk modeling, forecasting, marketing research, fraud detection, and risk detection and assessment. This research paper provides the evaluation of the Neural Network model and reflects it on the real-world data to provide optimum solutions for the problems with enhancement in the accuracy of results. This research discusses the appropriate use of hyperparameters like learning rate, batch size, optimizers, and activation functions like Sigmoid, ReLU, and Softmax to appropriately match a real-world problem. This research paper is a mere effort to evaluate the Neural Network model to better suit a real-world problem.

Keywords:

Artificial Intelligence, Neural Networks, Neural Network model, hyperparameters, activation function

Introduction

The world is being rapidly transformed by Artificial Intelligence (AI). Everything from autonomous vehicles to Internet of Things (IoT) connected devices have had AI play a major role in their development (NIST, 2021). Since the term intelligence has been the subject of much discussion and led to a lot of confusion, the precise definition of artificial intelligence has also posed a similar problem (Kok, et al., 2009). Artificial intelligence has been described as 'the ability of a digital computer or computer-controlled robot to perform tasks commonly associated with intelligent beings' (Copeland, 2020). In simple terms, it is the imitation of human behavior.

Machine learning is a branch of artificial intelligence that is focused on creating applications that optimize certain performance criteria by learning from data or experience without being programmed to do so (Alpaydin, 2014; IBM, 2020). An algorithm can be defined as a sequence of statistical processing steps or a sequence of instructions that can be carried out to convert the input to output (IBM, 2020: Alpaydin, 2014). These algorithms are 'trained' to identify patterns and characteristics present within large amounts of data to make forecasts and decisions based on new data (IBM, 2020). Data mining is defined as the application of machine learning methods to large databases (Alpaydin, 2014). In data mining, a simple model is generated from processing a huge volume of data and this model is then used for prediction for example (Alpaydin, 2014).

An artificial neural network (ANN), or simply neural network, is a computational learning system that is composed of thousands or even millions of simple processing nodes that emulate the biological neurons of animal brains

(Ladstätter & Garrosa, 2008; Hardesty, 2017; DeepAI, 2021). Machine learning algorithms make use of neural networks by which computers learn how to perform tasks through training examples (Hardesty, 2017; DeepAI, 2021).

A great many examples of machine learning surround us today. Spam detectors prevent spam emails to reach a user's inbox, websites recommend products based on what we bought previously, digital assistants respond to voice commands and are able to search the web and play music (IBM, 2020: Alpaydin, 2014). Finance banks build models by analyzing their past data to use in fraud detection, credit applications, and the stock market (Alpaydin, 2014). In healthcare, medical diagnosis makes use of learning programs; in telecommunications, network optimization and service improvement happen due to call pattern analysis (Alpaydin, 2014). Retail management, customer profiling, targeted marketing, market segmentation, etc. all make use of machine learning (Tzanis et al., 2006).

Related Works

Given the way machine learning is transforming nearly every domain in recent times, it is not surprising that machine learning algorithms are being tested and employed in a diverse number of environments and situations (Bhat & Ahmed, 2016; Muhsin et al., 2019; Almaqbali et al., 2020). Mupangwa et al. (2020) report that machine learning algorithms can be used to predict maize grain yields, although stating that some algorithms give more accurate predictions than others. Ladstätter and Garrosa (2008) applied different artificial neural network (ANN) architectures to their collected data in order to predict burnout among nursing professionals. Rashmi and Prashanth (2020) evaluated the effectiveness of different machine learning algorithms in their ability to predict bugs in the maintenance and development phase of the software development life cycle, concluding that artificial neural networks appeared to be the best model. Similarly, Hammouri et al. (2018) proposed a software bug prediction model using machine learning algorithms and reported that machine learning algorithms can be used to effectively predict future software bugs with a high accuracy rate.

Sentiment Analysis has been described as the most well-known branch of natural language processing (Tripathy, Agrawal & Rath, 2015). One definition of natural language processing is as follows: "Natural Language Processing is a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications." (Liddy, 2001, Introduction section, para. 2).

The intention of the author of a text can be analyzed using machine learning algorithms and classified and labeled as positive, neutral, and negative. This paper aims to compare the efficiency of neural networks in their ability to classify text data into such labels by using different activation functions and parameters. A test model (only including binary data) was developed using Python, and NumPy, a Python library, was connected to this model to apply different activation functions and parameters.

Neural Network Model comparison on activation functions and parameters

The efficiency of neural networks depends upon various factors like the selection of layers, hyperparameters, and neurons. It is also a fact that if the results reveal high accuracy on training (overfit) without validation then parameters can be decreased, however if there is low accuracy on training (underfit), then parameters need to be increased. We can opt. for automatically searching for the best hyperparameters with a grid search that includes learning rate, batch size, optimizer, dropout, etc. Activation functions on the other hand are very essential for introducing the non-linearity into our neural network data that contributes towards allowing more complex data to fit into a neural network model. The question arises which of the activation functions should be used. Figure 1 shows below different activation functions that are used.

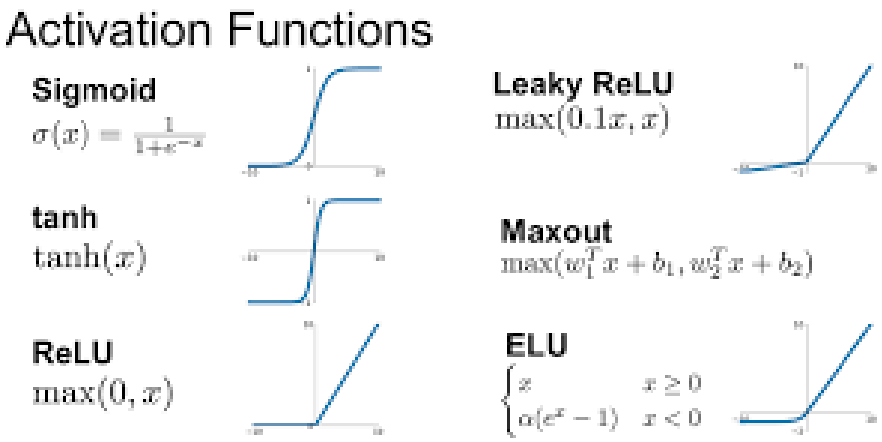


Figure 1 Activation Functions

Different activation functions are suitable for different classifications and situations, like ReLU avoids the vanishing gradient problem, Softmax function is good for single-label classification, and Sigmoid is a better choice for multi-label classification. To compare the efficiency of the models using a variety of parameters with different activation functions and different values in the parameters were used and the results of accuracy were compared in the neural network model. ReLU functions with the number of layers as 4 were used which obtained 76% accuracy. The model was trained using the liner data. Figure 2 shows the code and results that were achieved for the neural network model.

```
import numpy as np

train_df = pd.read_csv('./data/train.csv')
np.random.shuffle(train_df.values)

print(train_df.head())

model = keras.Sequential([
    keras.layers.Dense(4, input_shape=(2,), activation='relu'),
    keras.layers.Dense(2, activation='sigmoid')])

model.compile(optimizer='adam',
              loss=keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])

x = np.column_stack((train_df.x.values, train_df.y.values))

model.fit(x, train_df.color.values, batch_size=4)


2456/4000 [=================>............] - ETA: 0s - loss: 0.6487 - accuracy: 0.6792
2664/4000 [==================>...........] - ETA: 0s - loss: 0.6441 - accuracy: 0.6937
2872/4000 [====================>.........] - ETA: 0s - loss: 0.6387 - accuracy: 0.7100
3076/4000 [=====================>.......] - ETA: 0s - loss: 0.6329 - accuracy: 0.7230
3284/4000 [======================>......] - ETA: 0s - loss: 0.6278 - accuracy: 0.7323
3484/4000 [=======================>....] - ETA: 0s - loss: 0.6224 - accuracy: 0.7437
3720/4000 [========================>...] - ETA: 0s - loss: 0.6168 - accuracy: 0.7551
3980/4000 [========================>.] - ETA: 0s - loss: 0.6110 - accuracy: 0.7671
4000/4000 [==============================] - 1s 344us/sample - loss: 0.6105 - accuracy: 0.768
[Finished in 5.3s]
```

Figure 2. Results with 76% accuracy

Figure 2 indicates that for parameter value 4 with batch size 4 and the activation function as ReLU the accuracy achieved is 76%. The change in the parameters and batch size and introducing epochs will increase the accuracy to

almost 100% and the neural network can classify 100% of the training examples correctly. Figure 3 below shows the change in the parameters and the results that are achieved from the model. As is indicated from figure 3 that shows the results that are achieved from the neural network when the hyperparameters are slightly changed in the model, it has a significant impact on the overall results that are achieved. In the below diagram the epochs are given the value of 5 with the batch size 4 which indicates that as the model is having lesser data to process, its accuracy of classification increases with small data set compared to the large data set. It also indicates that the more numbers of periods/occurrences we train the data the accuracy increases to the maximum level. This difference can be easily seen between the results that are achieved from Figure 2 and Figure 3. The only difference between Figure 2 and Figure 3 is that, in the latter, only epochs are introduced with the value 5 that enhances the accuracy from 76% to 100%. The accuracy of training data enhances to 100% as the epochs are introduced without changing any other parameters and values or modifying the activation function. The results that were achieved while keeping the batch size 16 was almost 70%, however, when the batch size was decreased to 4, the accuracy of the results were enhanced by 6% and the overall accuracy received on the training data was 76%. The batch size can have a huge impact on the rate of accuracy, and, as the batch size increases, it can certainly have an impact on the accuracy of the data classification.

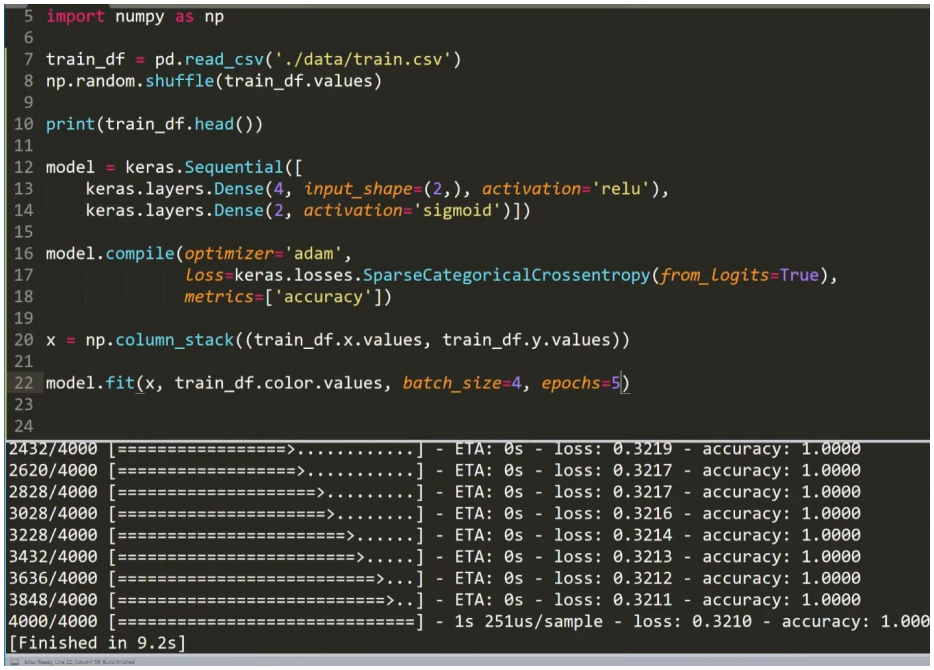
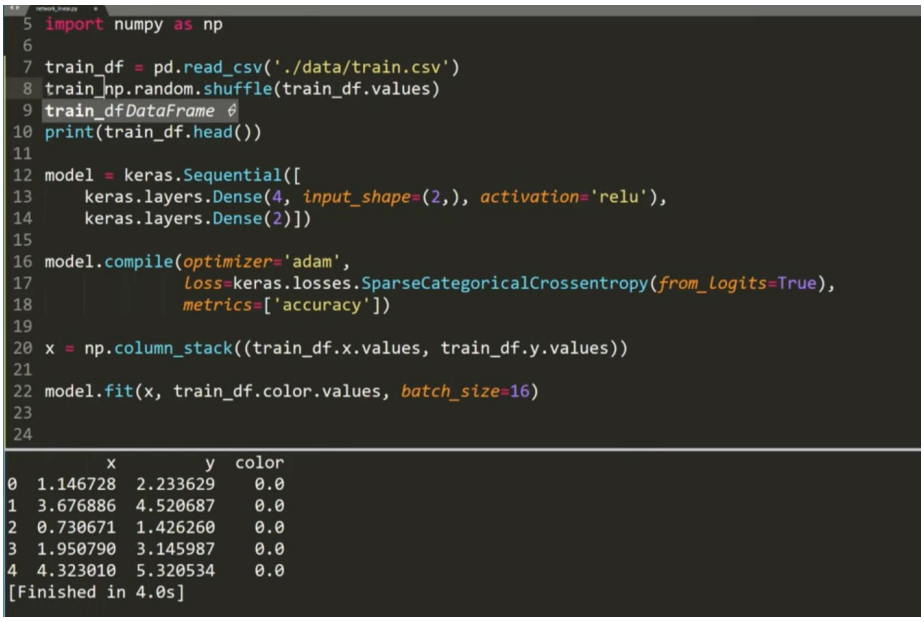


Figure 3 Results with 100% accuracy

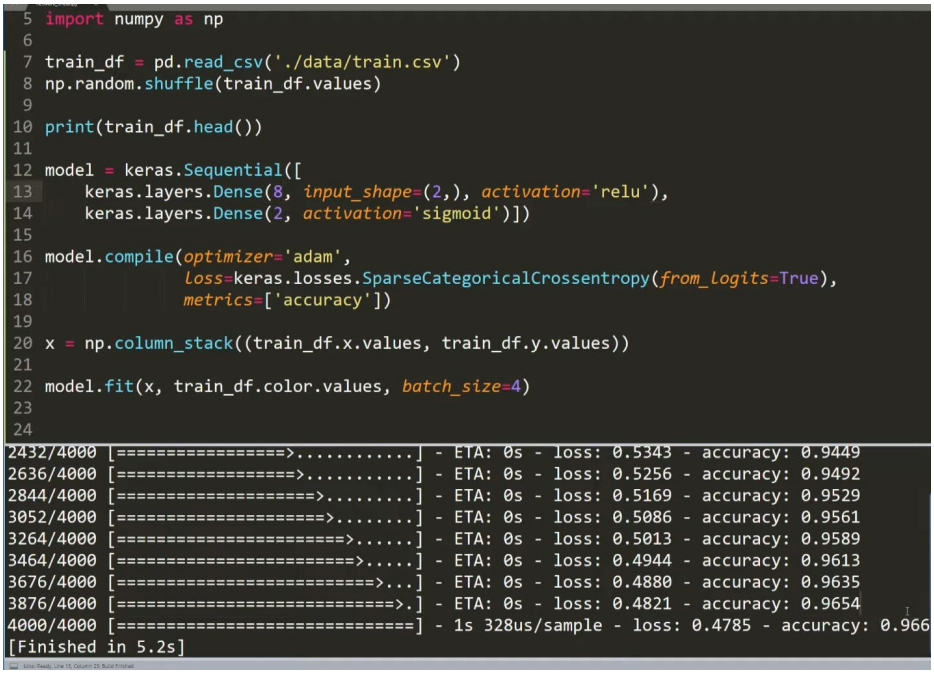
Now we shall concentrate on a different aspect of neural networks to find out the impact on the accuracy of classifying the data by the neural network. Keras represents the actual neural network model and offers two modes for the creation of the model. The one that is discussed in this research is the sequential API. A sequential model is suitable for the plain stack of layers where every layer has exactly one input tensor and one output tensor. We can access layers in the Keras by giving the code Keras. layers.Dense and then provide the values for various aspects. As we are concentrating on a fully connected feedforward network so the appropriate type to use in the Keras would be Dense. One of the important values that are passed to Dense would be the number of neurons on the first hidden layer and then we can pass the activation function which in our case first would be ReLU. The same is depicted in Figure 4 shown below.



```
import numpy as np

train_df = pd.read_csv('./data/train.csv')
train_np.random.shuffle(train_df.values)
train_dfDataFrame
print(train_df.head())

model = keras.Sequential([
    keras.layers.Dense(4, input_shape=(2,), activation='relu'),
    keras.layers.Dense(2)])

model.compile(optimizer='adam',
              loss=keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])

x = np.column_stack((train_df.x.values, train_df.y.values))

model.fit(x, train_df.color.values, batch_size=16)


          x         y  color
0  1.146728  2.233629    0.0
1  3.676886  4.520687    0.0
2  0.730671  1.426260    0.0
3  1.950790  3.145987    0.0
4  4.323010  5.320534    0.0
[Finished in 4.0s]
```

Figure 4 ReLU as activation function

It is ensured that training data is reshuffled, so the function np. random is used to shuffle the training data. We can further add the activation to the output data and it would be appropriate to add the sigmoid function as we are having binary classification. It is appropriate to use Sigmoid or Softmax for the binary classification. Figure 5 below provides the results if we are modifying the hyperparameter unit inside the Kera. This is clearly depicted in the below figure.



```
import numpy as np

train_df = pd.read_csv('./data/train.csv')
np.random.shuffle(train_df.values)

print(train_df.head())

model = keras.Sequential([
    keras.layers.Dense(8, input_shape=(2,), activation='relu'),
    keras.layers.Dense(2, activation='sigmoid')])

model.compile(optimizer='adam',
              loss=keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])

x = np.column_stack((train_df.x.values, train_df.y.values))

model.fit(x, train_df.color.values, batch_size=4)


2432/4000 [=================>............] - ETA: 0s - loss: 0.5343 - accuracy: 0.9449
2636/4000 [==================>...........] - ETA: 0s - loss: 0.5256 - accuracy: 0.9492
2844/4000 [====================>.........] - ETA: 0s - loss: 0.5169 - accuracy: 0.9529
3052/4000 [=====================>........] - ETA: 0s - loss: 0.5086 - accuracy: 0.9561
3264/4000 [======================>......] - ETA: 0s - loss: 0.5013 - accuracy: 0.9589
3464/4000 [=======================>.....] - ETA: 0s - loss: 0.4944 - accuracy: 0.9613
3676/4000 [========================>...] - ETA: 0s - loss: 0.4880 - accuracy: 0.9635
3876/4000 [=========================>.] - ETA: 0s - loss: 0.4821 - accuracy: 0.9654
4000/4000 [==============================] - 1s 328us/sample - loss: 0.4785 - accuracy: 0.966
[Finished in 5.2s]
```

Figure 5 Hyperparameter neuron units

As we can see, as the number of neurons on the training data is modified from 4 to 8, the accuracy rate is enhanced from 76% to 96%. The output layer is passed a sigmoid function and the batch size remains the same as shown in the above examples as 4.

Conclusion

The efficiency and accuracy of neural networks can be enhanced for classifying different types of data. Various aspects like hyperparameters, hyperparameter functions, batch size, number of neurons and iterations have a huge impact on the accuracy of the neural network. This research study shows the classification of the binary data and how various parameters can have a huge impact on the output and overall accuracy of the results. The research takes into consideration various scenarios in which the accuracy of the data may be impacted in a significant manner as there is a small modification on the values or other parameters. The accuracy of neural networks can be enhanced by opting for an appropriate activation function depending upon the characteristics of the data in hand, as well as by modifying other parameters like the number of neurons in the hidden layer, reputations, batch size of the data besides others. These parameters should be kept in consideration while designing a neural network model for a specific classification for enhanced efficiency and accuracy.

Limitations

This research study takes into consideration the binary data and the test is done on the data with changes in different parameters and activation functions which are appropriate for the binary data, however it may vary with multi-variable data and the results can be significantly different. This research study may not apply to a different type of data.

Acknowledgments

We are extremely thankful to Almighty Allah for bestowing us the patience, courage, and intellect to complete this research study. Our sincere gratitude to all the friends, colleagues, mentors, and contemporaries who lend a helping hand during the research study. We are extremely thankful to the faculty members and management of Middle East College for their support, encouragement, and guidance at all times. Appreciation is also due for our family members and parents without whose help guidance this work would not be possible.

References

Almaqbali, I. S. H., Al Khufairi, F. M. A., Khan, M. S., Bhat, A. Z., & Ahmed, I. (2020). Web Scrapping: Data Extraction from Websites. *Journal of Student Research*. https://doi.org/10.47611/jsr.vi.942

Alpaydin, E. (2014). *Introduction to Machine Learning*. [ProQuest Ebook Central version]. Retrieved from https://ebookcentral.proquest.com/lib/mecoman-ebooks/detail.action?docID=3339851

Bhat, A.Z. & Ahmed, I. (2016, May 3). *Big data for institutional planning, decision support and academic excellence* [Paper presentation]. 2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC), Muscat, Oman. https://doi.org/10.1109/ICBDSC.2016.7460353

Copeland, B. (2020, August 11). *Artificial intelligence*. *Encyclopedia Britannica*. https://www.britannica.com/technology/artificial-intelligence

Hammouri, A., Hammad, M., Alnabhan, M.M., & Alsarayrah, F. (2018). Software Bug Prediction using Machine Learning Approach. *International Journal of Advanced Computer Science and Applications, 9*. https://api.semanticscholar.org/CorpusID:3614956

Hardesty, L. (2017, April 14). Explained: Neural networks. *MIT News.* https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414

IBM. (2020, July 15). *Machine Learning.* https://www.ibm.com/cloud/learn/machine-learning

Kok, J. N., Boers, E.J.W., Kosters, W.A., Putten, P.V.D., & Poel, M. (2009). Artificial Intelligence: Definition, Trends, Techniques, And Cases. Paris: Eolss Publishers. Retrieved from http://www.eolss.net/sample-chapters/c15/e6-44.pdf

Ladstätter, F., & Garrosa, E. (2008). *Prediction of Burnout : An Artificial Neural Network Approach*. [ProQuest Ebook Central version]. Retrieved from https://ebookcentral.proquest.com/lib/mecoman-ebooks/detail.action?docID=594655

Liddy, E.D. (2001) Natural Language Processing. In *Encyclopedia of Library and Information Science* (2nd ed.). Retrieved from https://surface.syr.edu/cgi/viewcontent.cgi?referer=http://scholar.google.com/&httpsredir=1&article=1019&context=cnlp

Muhsin, T.F., Bhat, A.Z., Mohamed, I.A., & Khan, S. (2019, December 10) *Big Data Analytics for Enhancing Students Experience in Higher Education – A Case Study* [Paper presentation]. The 4th Middle East College Student Conference On Smart Technologies, Communication, Engineering, Archives and Management (STCEAM), Muscat, Oman. https://doi.org/10.47611/jsr.vi.949

Mupangwa, W., Chipindu, L., Nyagumbo, I., Mkuhlani, S., & Sisito, G. (2020). Evaluating machine learning algorithms for predicting maize yield under conservation agriculture in Eastern and Southern Africa. *SN Applied Sciences, 2*, 1-14. https://link.springer.com/article/10.1007%2Fs42452-020-2711-6

Neural Network. (2021, n.d.). DeepAI. https://deepai.org/machine-learning-glossary-and-terms/neural-network

NIST (2021, n.d.) *ARTIFICIAL INTELLIGENCE.* https://www.nist.gov/artificial-intelligence

Rashmi, P., & Kambli, P. (2020). Predicting Bug in a Software using ANN Based Machine Learning Techniques. *2020 IEEE International Conference for Innovation in Technology (INOCON)*, 1-5. https://doi.org/10.1109/INOCON50539.2020.9298203

Tripathy, A., Agrawal, A., & Rath, S. K. (2015). Classification of Sentimental Reviews Using Machine Learning Techniques. *Procedia Computer Science*, *57*, 821–829. https://doi.org/10.1016/j.procs.2015.07.523

Tzanis, G., Katakis, I., Partalas, I., & Vlahavas, I. (2006, July 10) *Modern Applications of Machine Learning* [Paper Presentation]. Proceedings of the 1st Annual SEERC Doctoral Student Conference (DSC 2006), Thessaloniki, Greece. https://www.researchgate.net/publication/228340464_Modern_Applications_of_Machine_Learning