International Journal of Academic Engineering Research (IJAER)
ISSN: 2643-9085
Vol. 7 Issue 5, May - 2023, Pages: 17-25

Earthquakes Prediction Using Deep Learning

Basel Y. El-Habil[1], Samy S. Abu-Naser[2]

Department of Information Technology,
Faculty of Engineering and Information Technology,
Al-Azhar University, Gaza, Palestine
[1]abunaser@alazhar.edu.ps

Abstract: *The goal of this project was to build deep learning model to predict earthquake location and magnitude and we also wanted to deploy the best model to predict whether there is an earthquake or not in year time. The model will be fast and more reliable earthquake detection as current methods depend on non-learning algorithms, so the data set we used contains 4,172 global maps generated from the data of the National Oceanic and Atmospheric Administration (NOAA), the earthquake data set includes Latitude, Longitude and Magnitude, from 2150 B.C.E. to 2022. Our deep learning model was trained and tested using the LSTM depending on the target the location and magnitude. We used the predication next farm architecture model for predicting earthquakes architecture where the input is 64x64 images (maps). There's one convolutional layer a max pooling layer drop out layer flattening layer and three dense layers so the second type of model. We used LSTM and the time series data to see if we could remove the need to generate images as was needed for the CNN so the model was trained and tested on the raw data. It is a map earthquake shown here and the black circle here shows the predicated earthquake, so the LSTM architecture input the seismic maps then there was a simple RNN layer an LSTM layer and dropout layer. Another LSTM layer and two dense layers so for the results here is a comparison of the confusion matrixes for the CNN versus the LSTM they both had good accuracy precision and recall but the RNN was a little better with 98 to 99 accuracy precision and recall versus about 99 so for the S wave arrival time prediction. There are some more results this is graphs of the predicted value versus the observed value for the RNN and for the LSTM a perfect prediction is indicated by black circles and you can see that the mean error value for the RNN is quite a bit lower than the loss error value for the LSTM. The RNN was the best model and overall here is a map of the four targets that we tried to predict which was classifying earthquakes versus noise predicting magnitude, location and time arrivals. We convert the earthquakes data to maps and sent this to one npy file. The python function runs the model and predicts the image automatically.*

Keywords: Artificial Intelligence, Neural Network, Deep Learning, Earthquakes Prediction.

1-Introduction

1.1 Deep Learning

Deep learning is a new type of data-driven AI. It stacks multiple layers of machine-learning models, one on top of the other. This allows it to learn more complex inputs and produce more complex outputs. We typically create deep-learning models using a neural network. A neural network is a graph of nodes and edges based roughly on the organization of neurons in an organic brain. The nodes represent neurons in a brain [1-5].

The edges represent the connections between the neurons. First, we feed data into the neural network via its input neurons. Next, mathematical operations are performed on the data in each of the neurons. Then, each neuron forwards its resulting value to all of the other neurons that it's connected to. We repeat this process for all of the nodes in the hidden layer of the networks well as the edges in the hidden layer. Finally, the network produces a prediction from its output neurons [6,7].

There's a bit of math involved to make this entire process work. However, we're going to skip over all the math to keep things simple. A deep neural network is a neural network with more than one hidden layer. Adding more hidden layers allows the network to model progressively more complex functions. For example, imagine we want to teach a deep neural network how to detect human faces [8-10].

First, we would feed a set of labeled images into the input layer of this network. We do this to teach the network the faces of each person and their corresponding name. The first layer of the neural network would learn to detect geometric primitives. For example: horizontal, vertical, and diagonal lines. The second hidden layer would learn to detect more complex facial features. For example: Eyes, noses, and mouths. The third hidden layer would learn to detect the general pattern for entire faces. The output layer would detect the most abstract representation of a person. In this case, the name of the person being recognized. Each layer learns to extract more complex features from the preceding layer. As a result, the data becomes more abstract with each additional layer in the network [11-15].

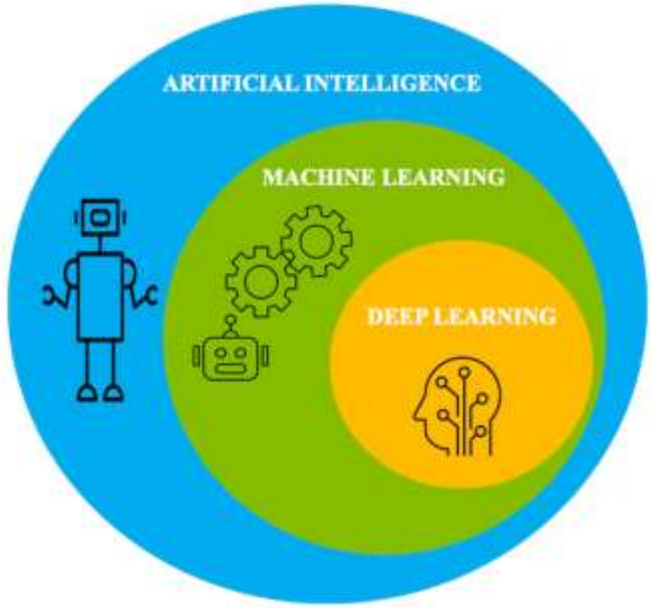

Figure 1: Deep Learning [16]

1.2 Earthquakes

Our planet appears eerily still, but every mountain range and every chasm on its face is a scar with many telling a story of when the earth rumbled to life. Earthquakes occur around the world, they've been recorded on all seven continents, but most quakes take place in just three regions. The mid-Atlantic ridge an underwater line that runs down the Atlantic Ocean. The alpide belt which stretches from the Mediterranean to Southeast Asia and the circum-pacific belt which traces along the edges of the Pacific Ocean and is where about 80% of all earthquakes occur. These areas experience the most earthquakes due to what lies beneath the surface. Earthquakes are the result of pressurespecifically pressure caused by extreme stress in the Earth's crust. That stress can be caused by volcanic activity or even man-made activities in certain areas. However, most earthquake inducing stresses caused by the movement of tectonic plates. Tectonic plates are constantly moving either against away along or underneath each other, but sometimes their edges may catch and stick the plates. However, continue to move or at least attempt to energy from this attempted movement fields around the edges. Sticking point creating immense pressure until the edges are forced to let go and the plates slip. This causes a sudden and powerful release of energy so powerful that it breaks the Earth’s crust. This fracturing emits shockwaves through the ground and causes intense vibrations or quakes [17-22].

In fact, the world’s most earthquake-prone regions are where the most geologically active plates meetearthquakes or any seismic activity are recorded by seismographs. When the ground shakes seismographs oscillate drawing a jagged line to reflect this movement. The more extreme the earthquake of the jagged line these recorded motions are then used to measure the earthquake strength or magnitude. While several scales of magnitude exist the one seismologist prefer is the moment magnitude scale. It has no upper limit and it measures earthquakes logarithmically. This means that each magnitude on its scale is ten times greater than the one before. It unlike the now rarely used Richter scale the moment magnitude scale can be applied globally and can measure quakes of the highest magnitudes. The largest recorded earthquake occurred near Bolivia Chile in 1960 nestled within the circum-pacific belt. The Valdivia earthquake was the most powerful in a series of quakes that struck. The region measuring at a magnitude of about 9.5 in addition to causing devastating tremors on land. The earthquake also generated a deadly tsunami reaching up to 80 feet high. The tsunami raced across the Pacific Ocean hitting faraway countries like the Philippines and Japan in fact data from seismographs showed that the shock wavesemitted by the Valdivia earthquake continued to shake the entire planet for days. Some earthquake prone areas have adapted various ways to protect their communities. Buildings and bridges are designed to sway rather than break when an earthquake occurs. The public is educated on how to protect themselves during a seismic event and government officials enact drills to ensure the protection of their people. Earthquakes can leave behind incredible devastation but these same forces have also created magnificent features with each adding character to a planet so unique [23-26].

2-STUDY OBJECTIVES

1) Develop an unsupervised (semi-supervised) model for Earthquakes prediction model.

2) Present a new Long Short-Term Memory LSTM-based learning approach to solve the random weight initialization problem of Deep Long Short-Term Memory DLSTM.
3) Propose a robust forecasting application based on the time series maps, which could be used to convert those observations into patterns that can be utilized easily for future projections.
4) Help the scientists and policymakers, in their resolving to act and consider potential benefits of reducing the Earthquakes crisis.

3- Recurrent Neural Networks (RNN)

The convolutional network uses shared parameters across space to extract patterns over an image. Recurrent Neural Networks (RNN) do the same thing but over time instead of space. This is the idea behind recurrent neural networks. Imagine that we have a sequence of events at each point in time you want to decide about what's happened so far in the sequence. Let that you have a sequence of events at each point in time. We want to make a decision about what's happened so far in the sequence. If your sequence is reasonably stationary, we can use the same classifier at each point in time. That simplifies things a lot already but since this is a sequence you also want to consider the past, everything that happened before that point. When natural thing to do here is to use the state of the previous classifier. What happened before recursively now we would need a very deep neural network to remember far in the past. This sequence could have hundreds thousands of steps it would basically mean to have a deep network with hundreds or thousands of layers, but instead we're going to use tying again and have a single model responsible for summarizing the past and providing that information to your classifier what we end up with is a network. With a relatively simple repeating pattern with part of your classifier connecting to the input at each time, step and another part called the recurrent connection connecting you to the past at each step [14].

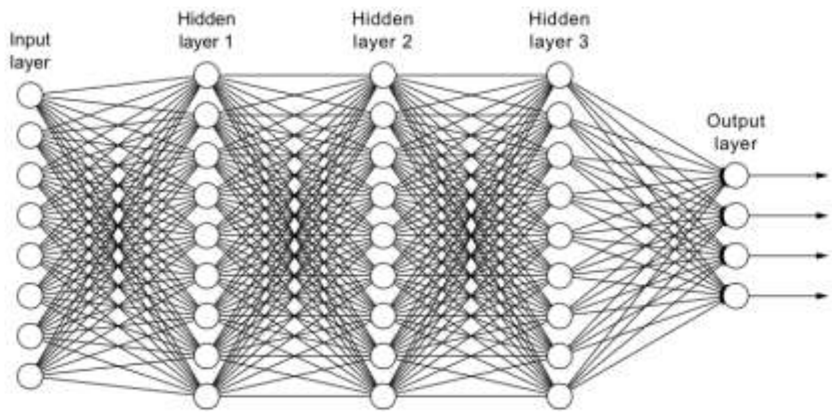

Figure 2 : Architecture of Neural Networks (Chollet, 2021)

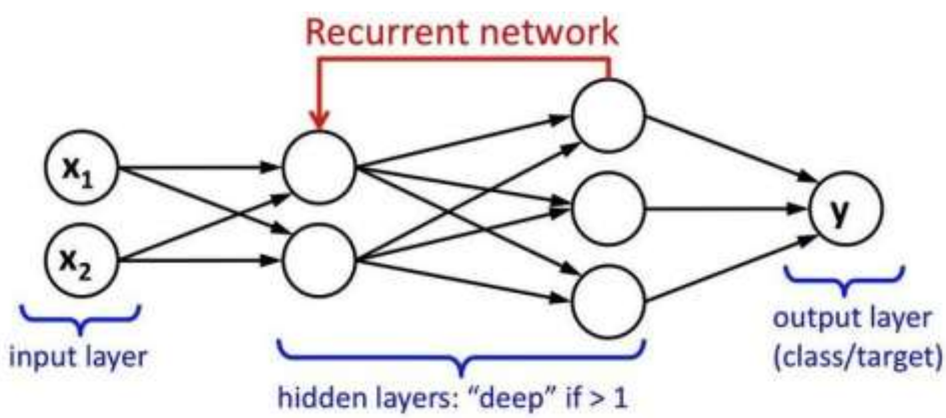

Figure 3 : Recurrent Neural Networks [27-30]

3.1 Long Short-Term Memory Networks

LSTM stands for long short-term memory, it looks like with the help of diagrams here's how a standard recurrent neural network can be represented. We have passed the input xt along with the output from the previous node, which is h-1 to this central node then apply the 10h activation function and generate the output which is hg. Now here's how we can present the LSTM recurrent neural

International Journal of Academic Engineering Research (IJAER)
ISSN: 2643-9085
Vol. 7 Issue 5, May - 2023, Pages: 17-25

network; its repeating module has a bit different structure. Instead of having a single neural network layer four interacting layers are communicating extraordinary. This complete box is called the LSTM cell it has three main parts: the cell state, the hidden state and the gates like forgot which also known as remember input and output gate [31-35].

The Long Short-Term Memory (LSTM) network is a type of artificial neural network that can process and classify sequential data. It is commonly used in natural language processing, speech recognition, and time series prediction. The LSTM network is able to overcome the vanishing gradient problem by allowing gradients to flow through a long chain of memory cells, which makes it particularly well-suited for processing sequences with long-term dependencies. This is accomplished by selectively forgetting or retaining information from previous time steps, as well as storing new information in the memory cells [36-40].

We have three sigmoid components, so let's expand Figure 4, we can see that the information is going in a straight line right there and we call this the cell state which means the state of the cell at a certain state. We are multiplying whatever this value with the state value there, so the cell state is modified if that value is 1 then we are not modifying the state value. The hidden state is meant to encode a kind of characterization of the previous time steps data. It is important to note that the hidden state doesn't equal the output or the prediction. It is merely an encoding of the most recent time steps that said the hidden state at any point can be processed to obtain more meaningful data, in other words the cell state act as a transport highway that transfers relative information all the way down. The sequence chain you can think of it as the memory of the network. The cell state can carry relevant information throughout the processing of the sequence. So, even information from the earlier time steps can make its way to the later time steps reducing the effect of short-term memory. As the cell state goes on its journey information gets added or removed to the cell state via the gates the gates are different neural networks that decide which information is allowed on the cell state, in other words the gates can learn what information is relevant to keep or forget during training. We have three different goods that regulate information flow in an LSTM cell: the first one is the forget git as you can see the highlighted portion of the diagram this gate decides what information should be thrown away or kept information from the previous hidden state, which we called the h_{t-1} and the information from the current input which we called the x_t is passed through the sigmoid function values come out between 0 and 1. The closer to 0 means to forget and the closer to 1 means to keep and that's the reason we are not modifying the state of the cell when this value is 1. To update the cell state we have the input gate in this layer there are two parts one is the sigmoid function and the other one is the 10 h function in the sigmoid function it decides which value to let through by transforming the values to be between 0 and 1. 0 means not important and 1 means important than the 10 h function gives weightage to the values which are passed deciding their level of importance by squishing them between -1 and 1. Then we multiply the 10 h output with the sigmoid output the sigmoid output will decide which information is important to keep from the 10 h output now. At this stage we have enough information to calculate the cell state. We will utilize the output from the forget gate and from the input gate to perform this calculation, which gives us a new cell state next. We have the output gate which decides what the next hidden state should be. Remember that the hidden state contains information on previous inputs the hidden state is also used for predictions. First we pass the previous hidden state which is h_{t-1} and the current input which is x_t into a sigmoid function then we pass the newly modified cell state to the 10 h function we multiply the tanh output with the sigmoid output to decide what information. The hidden state should carry the output is the hidden state, the new cell state and the new hidden state are then carried over to the next time step. In simple words here the sigmoid layer decides which part of the cell state will be present in the output whereas 10h layer shifts the output in the range of -1 to 1. that's how an LSTM recurrent neural network performs the learning while taking the long context into the account [41-45].

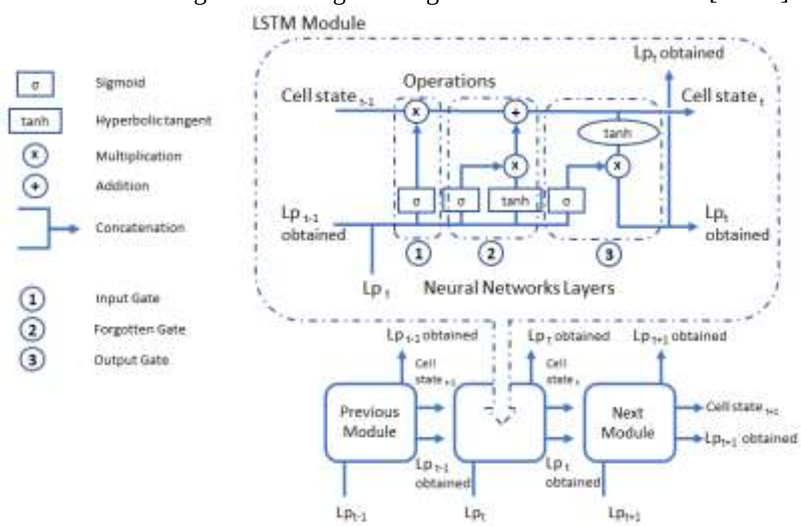

Figure 4 : General scheme of an Long Short-Term Memory neural networks (LSTM) [46-49]

4- Methodology

4.1 Dataset

The dataset for training, Validation and testing of algorithms for Earthquakes Prediction was collected form National Centers for Environmental Information website. The dataset has 4,172 Earthquakes maps from 2150 B.C.E. to 2022[1].
We used 90% of the data for training, and 10% for the testing, while the whole dataset for validation, and each record was the yearly earthquakes all over the world disaggregated by level, latitudes and longitudes.

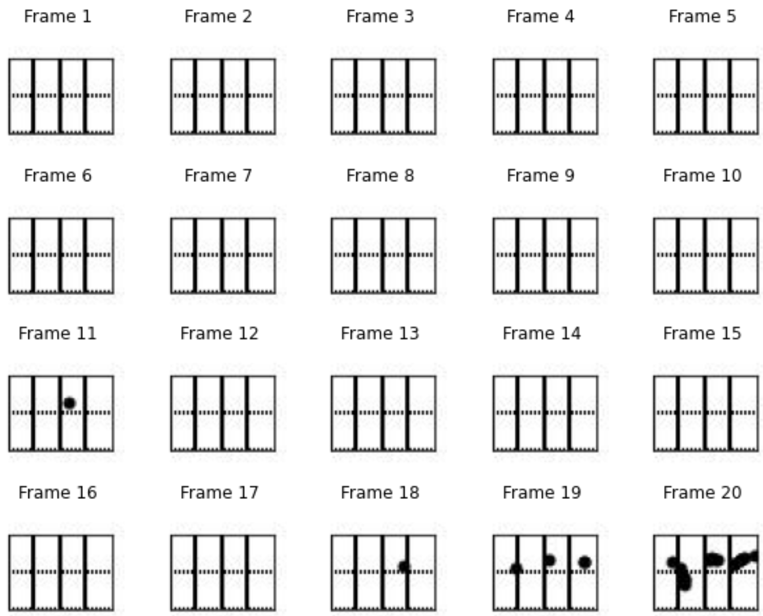

Figure 5 : Samples from Dataset

4.2 Language and tools used

Python is considered one of the most famous and most popular languages that support machine learning with the Google Colab environment, because it contains supportive libraries, is flexible, easy to use and open source, and it has the ability to train and validate very fast.
We used the following libraries for developing the new proposed model:

- Pandas - Data Structure and Analysis.
- Keras - in deep learning. It wraps the efficient numerical computation libraries Theano and TensorFlow(Brownlee, 2020).
- Matplotlib - used for graphs and charts.
- Geographic Data with Basemap- used for generate the maps from netcdf data.

4.3 Network Architecture

In Figure 6, the Model Architecture which contains from 6 layers and Input layer

[1] https://www.ngdc.noaa.gov/hazel/view/hazards/earthquake/event-data

International Journal of Academic Engineering Research (IJAER)
ISSN: 2643-9085
Vol. 7 Issue 5, May - 2023, Pages: 17-25

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, None, 64, 64, 1)  0
                             ]

 conv_lstm2d (ConvLSTM2D)    (None, None, 64, 64, 64)  416256

 batch_normalization (BatchN  (None, None, 64, 64, 64)  256
 ormalization)

 conv_lstm2d_1 (ConvLSTM2D)  (None, None, 64, 64, 64)  295168

 batch_normalization_1 (Batc  (None, None, 64, 64, 64)  256
 hNormalization)

 conv_lstm2d_2 (ConvLSTM2D)  (None, None, 64, 64, 64)  33024

 conv3d (Conv3D)             (None, None, 64, 64, 1)   1729

=================================================================
Total params: 746,689
Trainable params: 746,433
Non-trainable params: 256
_________________________________________________________________
```

Figure 6 : Model Architecture

In the Figure 7, we illustrate the last 40 epochs of the training and validation of the proposed model.

```
Epoch 1/40
38/38 [==============================] - 30s 801ms/step - loss: 0.0468 - acc: 0.7114 - val_loss: 0.0595 - val_acc: 0.7100 - lr: 0.0010
Epoch 2/40
38/38 [==============================] - 31s 820ms/step - loss: 0.0467 - acc: 0.7114 - val_loss: 0.0532 - val_acc: 0.7109 - lr: 0.0010
Epoch 3/40
38/38 [==============================] - 31s 815ms/step - loss: 0.0466 - acc: 0.7114 - val_loss: 0.0486 - val_acc: 0.7110 - lr: 0.0010
Epoch 4/40
38/38 [==============================] - 31s 809ms/step - loss: 0.0465 - acc: 0.7114 - val_loss: 0.0504 - val_acc: 0.7108 - lr: 0.0010
Epoch 5/40
38/38 [==============================] - 31s 815ms/step - loss: 0.0465 - acc: 0.7114 - val_loss: 0.0494 - val_acc: 0.7110 - lr: 0.0010
Epoch 6/40
38/38 [==============================] - 31s 812ms/step - loss: 0.0465 - acc: 0.7114 - val_loss: 0.0475 - val_acc: 0.7110 - lr: 0.0010
Epoch 7/40
38/38 [==============================] - 31s 812ms/step - loss: 0.0465 - acc: 0.7114 - val_loss: 0.0496 - val_acc: 0.7109 - lr: 0.0010
Epoch 8/40
38/38 [==============================] - 31s 813ms/step - loss: 0.0464 - acc: 0.7114 - val_loss: 0.0474 - val_acc: 0.7110 - lr: 0.0010
Epoch 9/40
38/38 [==============================] - 31s 814ms/step - loss: 0.0462 - acc: 0.7114 - val_loss: 0.0474 - val_acc: 0.7110 - lr: 0.0010
Epoch 10/40
38/38 [==============================] - 31s 814ms/step - loss: 0.0463 - acc: 0.7114 - val_loss: 0.0483 - val_acc: 0.7109 - lr: 0.0010
Epoch 11/40
38/38 [==============================] - 31s 814ms/step - loss: 0.0461 - acc: 0.7114 - val_loss: 0.0485 - val_acc: 0.7110 - lr: 0.0010
Epoch 12/40
38/38 [==============================] - 31s 813ms/step - loss: 0.0461 - acc: 0.7114 - val_loss: 0.0468 - val_acc: 0.7110 - lr: 0.0010
Epoch 13/40
38/38 [==============================] - 31s 814ms/step - loss: 0.0460 - acc: 0.7114 - val_loss: 0.0469 - val_acc: 0.7110 - lr: 0.0010
Epoch 14/40
38/38 [==============================] - 31s 813ms/step - loss: 0.0460 - acc: 0.7114 - val_loss: 0.0467 - val_acc: 0.7110 - lr: 0.0010
Epoch 15/40
38/38 [==============================] - 31s 813ms/step - loss: 0.0460 - acc: 0.7114 - val_loss: 0.0466 - val_acc: 0.7110 - lr: 0.0010
Epoch 16/40
38/38 [==============================] - 31s 813ms/step - loss: 0.0460 - acc: 0.7114 - val_loss: 0.0465 - val_acc: 0.7110 - lr: 0.0010
Epoch 17/40
38/38 [==============================] - 31s 813ms/step - loss: 0.0460 - acc: 0.7114 - val_loss: 0.0465 - val_acc: 0.7110 - lr: 0.0010
Epoch 18/40
38/38 [==============================] - 31s 813ms/step - loss: 0.0459 - acc: 0.7114 - val_loss: 0.0465 - val_acc: 0.7110 - lr: 0.0010
Epoch 19/40
38/38 [==============================] - 31s 814ms/step - loss: 0.0459 - acc: 0.7114 - val_loss: 0.0467 - val_acc: 0.7110 - lr: 0.0010
Epoch 20/40
38/38 [==============================] - 31s 814ms/step - loss: 0.0459 - acc: 0.7114 - val_loss: 0.0469 - val_acc: 0.7110 - lr: 0.0010
Epoch 21/40
38/38 [==============================] - 31s 813ms/step - loss: 0.0459 - acc: 0.7114 - val_loss: 0.0467 - val_acc: 0.7110 - lr: 0.0010
```

Figure 7 : The model starts training and logs the loss and accuracy

While training the model, the loss and accuracy metrics are shown. This model achieves training and validation accuracy of about 71%, and training and validation loss is 0.04.6

International Journal of Academic Engineering Research (IJAER)
ISSN: 2643-9085
Vol. 7 Issue 5, May - 2023, Pages: 17-25

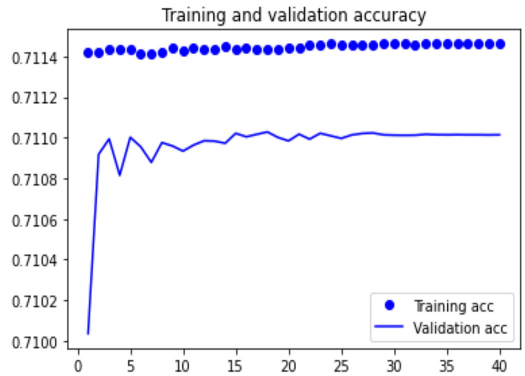

Figure 8 : Training and Validation accuracy , 40 epochs

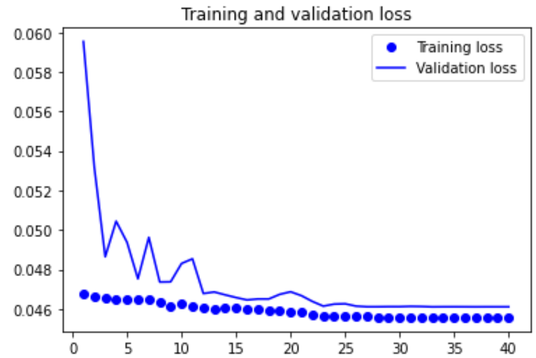

Figure 9 : Training and Validation loss, 40 epochs

During the training of the model the loss and accuracy values are recorded, and a validation dataset was used to check the performance of the model. Also, "Matplotlib" was used to draw a chart showing the tracking of the results of the training and validation process in terms of loss and accuracy, as shown in the Figure 8 and Figure 9.
We notice that the loss of validation and training of decreases with each iteration, and this is a good thing, and it gives us an indication that the model achieves its goals. On the other hand, it appears that the validation accuracy of the model increases to reach 71%, which is an acceptable percentage.

5-Conclusion

The results were summarized by displaying them in table for ease of comparison in terms of accuracy and loss. Because the amount of data used is large, and our model gave the acceptable accuracy rate of 71%.

References

1. Abunasser, B. S., AL-Hiealy, M. R. J., Zaqout, I. S. and Abu-Naser, S. S. "Breast Cancer Detection and Classification using Deep Learning Xception Algorithm" International Journal of Advanced Computer Science and Applications(IJACSA), 13(7), 223-228, 2022. http://dx.doi.org/10.14569/IJACSA.2022.0130729
2. Abunasser, B.S., AL-Hiealy, M.R. J., Barhoom, A. M. Almasri A. R. and Abu-Naser, S. S. "Prediction of Instructor Performance using Machine and Deep Learning Techniques" International Journal of Advanced Computer Science and Applications(IJACSA), 13(7), 78-83, 2022. http://dx.doi.org/10.14569/IJACSA.2022.0130711
3. Taha, A. M. H., Ariffin, D. S. B. B., Abu-Naser, S. S. A Systematic Literature Review of Deep and Machine Learning Algorithms in Brain Tumor and Meta-Analysis, Journal of Theoretical and Applied Information Technology, 2023, Vol. 101. No. 1, pp. 21-36.
4. Alkayyali, Z. K. D., Idris, S. A. B, Abu-Naser, S. S. A Systematic Literature Review of Deep and Machine Learning Algorithms in Cardiovascular Diseases Diagnosis, Journal of Theoretical and Applied Information Technology, 2023, Vol. 101. No. 4.
5. amazon.science/blog/new-alexa-features-interactive-teaching-by-customers last visited 20/1/2023
6. Zarandah, Q. M. M., Daud, S. M., Abu-Naser, S. S. A Systematic Literature Review Of Machine and Deep Learning-Based Detection And Classification Methods for Diseases Related To the Respiratory System, Journal of Theoretical and Applied Information Technology, 2023, Vol. 101. No. 4.
7. Abunasser, B. S. Daud,S. M., Zaqout,I., Abu-Naser S. S. Convolution Neural Network For Breast Cancer Detection And Classification - Final Results. Journal of Theoretical and Applied Information Technology, 2023, Vol. 101. No. 1, pp. 315-329.
8. Almasri, A.R., Yahaya, N.A., Abu-Naser, S.S. Instructor Performance Modeling For Predicting Student Satisfaction Using Machine Learning - Preliminary Results. Journal of Theoretical and Applied Information Technology, 2022, 100(19), pp. 5481–5496
9. Arqawi, S.M., Abu Rumman, M.A., Zitawi, E.A., ...Abunasser, B.S., Abu-Naser, S.S. Predicting Employee Attrition And Performance Using Deep Learning. Journal of Theoretical and Applied Information Technology, 2022, 100(21), pp. 6526–6536
10. Arqawi, S.M., Zitawi, E.A., Rabaya, A.H., Abunasser, B.S., Abu-Naser, S.S., "Predicting University Student Retention using Artificial Intelligence", International Journal of Advanced Computer Science and Applications , 2022, vol. 13, no. 9, pp. 315–324
11. Barhoom, A.M.A., Al-Hiealy, M.R.J., Abu-Naser, S.S. Bone Abnormalities Detection and Classification Using Deep Learning-VGG16 Algorithm. Journal of Theoretical and Applied Information Technology, 2022, 100(20), pp. 6173–6184
12. Barhoom, A.M.A., Al-Hiealy, M.R.J., Abu-Naser, S.S. Deep Learning-Xception Algorithm for Upper Bone Abnormalities Classification. Journal of Theoretical and Applied Information Technology, 2022, 100(23), pp. 6986–6997
13. El-Habil, B.Y., Abu-Naser, S.S. Global Climate Prediction Using Deep Learning. Journal of Theoretical and Applied Information Technology, 2022, 100(24), pp. 4824–4838
14. Alzamily, J. Y. I., Ariffin, S. B., Abu-Naser, S. S. Classification of Encrypted Images Using Deep Learning –Resnet50. Journal of Theoretical and Applied Information Technology, 2022, 100(21), pp. 6610–6620
15. Saleh, A., Sukaik, R., Abu-Naser, S.S. Brain tumor classification using deep learning. Proceedings - 2020 International Conference on Assistive and Rehabilitation Technologies, iCareTech 2020, 2020, pp. 131–136, 9328072
16. Abunasser, B., AL-Hiealy, M. R., Zaqout, I., Abu-Naser, S. Convolution Neural Network for Breast Cancer Detection and Classification Using Deep Learning. Asian Pacific Journal of Cancer Prevention, 2023; vol. 24, no. 2, pp. 531-544. doi: 10.31557/APJCP.2023.24.2.531
17. Ayodele, T. O. (2010). Types of machine learning algorithms. New advances in machine learning, 3, 19-48.
18. Brownlee, J. (2020). Your First Deep Learning Project in Python with Keras Step-By-Step. Machine Learning Mastery.
19. Campos-Taberner, M., García-Haro, F. J., Martínez, B., Izquierdo-Verdiguier, E., Atzberger, C., Camps-Valls, G., & Gilabert, M. A. (2020). Understanding deep learning in land use classification based on Sentinel-2 time series. Scientific reports, 10(1), 1-12.
20. Chattopadhyay, R., Vintzileos, A., & Zhang, C. (2013). A description of the Madden–Julian oscillation based on a self-organizing map. Journal of climate, 26(5), 1716-1732.
21. Chollet, F. (2021). Deep learning with Python: Simon and Schuster.
22. DiPietro, R., & Hager, G. D. (2020). Deep learning: RNNs and LSTM. In Handbook of medical image computing and computer assisted intervention (pp. 503-519): Elsevier.
23. diplomacy, T. c. (2018). Making Development Sustainable through Climate Action.
24. Donges, N. (2021). A guide to rnn: Understanding recurrent neural networks and lstm networks. web resource: https://builtin. com/data-science/recurrent-neural-networksand-lstm.

25. Dueben, P. D., & Bauer, P. (2018). Challenges and design choices for global weather and climate models based on machine learning. Geoscientific Model Development, 11(10), 3999-4009.
26. Eckstein, D., Künzel, V., & Schäfer, L. (2021). Global climate risk index 2021. Who Suffers Most from Extreme Weather Events, 2000-2019.
27. Gorricha, J., Lobo, V., & Costa, A. C. (2013). A imagework for exploratory analysis of extreme weather events using geostatistical procedures and 3D self-organizing maps. International Journal on Advances in Intelligent Systems, 5(NA), 16-26.
28. Guo, T., Dong, J., Li, H., & Gao, Y. (2017). Simple convolutional neural network on image classification. Paper presented at the 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA).
29. Heidenreich, H. (2018). The Future with Reinforcement Learning.
30. Iglesias, G., Kale, D. C., & Liu, Y. (2015). An examination of deep learning for extreme climate pattern analysis. Paper presented at the The 5th International Workshop on Climate Informatics.
31. Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. Science, 349(6245), 255-260.
32. Leetaru, K. (2019). Why Machine Learning Needs Semantics Not Just Statistics, Forbes. In.
33. Liu, Y., Racah, E., Correa, J., Khosrowshahi, A., Lavers, D., Kunkel, K., . . . Collins, W. (2016). Application of deep convolutional neural networks for detecting extreme weather in climate datasets. arXiv preprint arXiv:1605.01156.
34. Moawad, A. (2018). Neural networks and back-propagation explained in a simple way. Medium [online]. Feb, 1.
35. Nasa, E. (2019). World of Change: Global Temperatures. In: Retrieved from Earth Observatory Nasa: https://earthobservatory. nasa. gov ….
36. Navarro, J. M., Martínez-España, R., Bueno-Crespo, A., Martínez, R., & Cecilia, J. M. (2020). Sound levels forecasting in an acoustic sensor network using a deep neural network. Sensors, 20(3), 903.
37. Parrish, K. (2018). Deep learning vs. machine learning: what's the difference between the two. Online: https://www. digitaltrends. com/cool-tech/deep-learning-vs-machine-learning-explained/2/[Last accessed: 08.05. 2018].
38. Pendergrass, A. G., Meehl, G. A., Pulwarty, R., Hobbins, M., Hoell, A., AghaKouchak, A., . . . Hoffmann, D. (2020). Flash droughts present a new challenge for subseasonal-to-seasonal prediction. Nature Climate Change, 10(3), 191-199.
39. Phi, M. (2018). Illustrated Guide to Recurrent Neural Networks. In.
40. Pritt, M., & Chern, G. (2017). Satellite image classification with deep learning. Paper presented at the 2017 IEEE Applied Imagery Pattern Recognition Workshop (AIPR).
41. Rivera, A. T. (2020). Machine Learning Basics | What is Machine Learning? | Introduction to Machine Learning.
42. Rushingabigwi, G., Zhang, J., Bachagha, T., Kalisa, W., Henchiri, M., Shahzad, A., . . . Bugingo, C. N. (2018). The Influence of Dust and Black Carbon on Clouds, in Africa. Journal of Computer and Communications, 6(11), 342-352.
43. Saravia, E. (2018). Deep Learning for NLP: An Overview of Recent Trends. Retrieved November, 27, 2018.
44. Shaheen, A., Waheed, U. b., Fehler, M., Sokol, L., & Hanafy, S. (2021). GroningenNet: Deep Learning for Low-Magnitude Earthquake Detection on a Multi-Level Sensor Network. Sensors, 21(23), 8080.
45. Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., & Woo, W.-c. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. Advances in neural information processing systems, 28.
46. Shin, T. (2020). A Beginner-Friendly Explanation of How Neural Networks Work.
47. United Nations, D. o. E. a. S. A. (2018). 68% of the world population projected to live in urban areas by 2050, says UN. In DESA (Ed.).
48. Weyn, J. A., Durran, D. R., & Caruana, R. (2020). Improving data-driven global weather prediction using deep convolutional neural networks on a cubed sphere. Journal of Advances in Modeling Earth Systems, 12(9), e2020MS002109.
49. WMO. (2021). United in Science 2021. Retrieved from