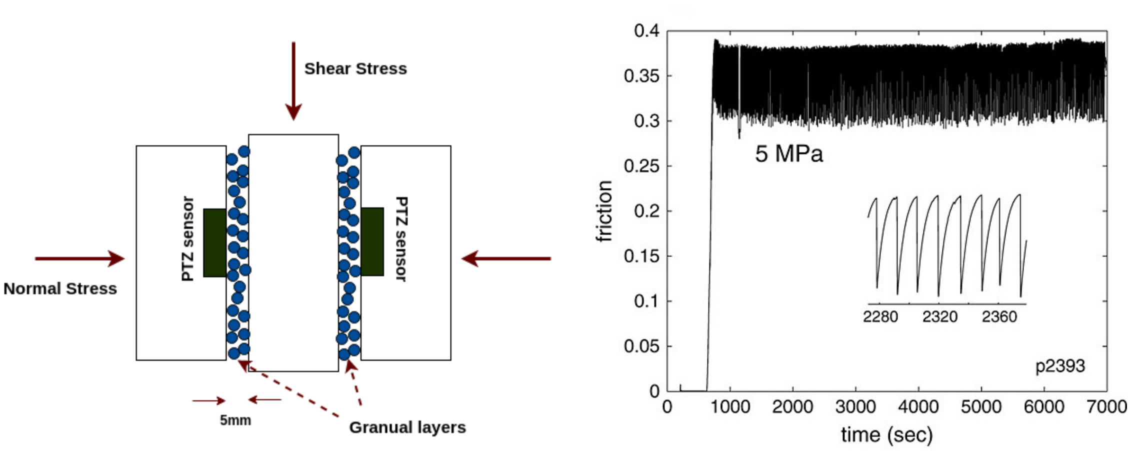
Predicting the time left to Earthquake using Deep Learning Models

Abstract. Predicting the earthquake is an important problem in the earth science community. Several models have been proposed to predict the time to next earthquake. However, all those methods are based on traditional machine learning algorithms. Recently, deep learning based models have obtained breakthrough achievements over traditional machine learning models in various data processing tasks. Hence, in this paper, we propose and analyze different deep learning based models to predict the time to the next earthquake. From our knowledge, this is the first work in which deep learning is used to predict the earthquakes. We perform several experiments on the proposed models and compare them in terms of the mean absolute error (MAE) measurement. From the different experiments, we found that the GRU-Conv1D model obtains better MAE value than that of other models.

Keywords: Deep Learning, LSTM, GRU, Conv-GRU, GRU-Conv

1. Inroduction

Due to the destructive nature of earthquakes, predicting its occurence is an important task to the earth science community. There are mainly three parameters used in an earthquake forecast. Firstly when it will occur, secondly the magnitude of the earthquake and thirdly where it has ocurred. In this manuscript, the focus is on the former; predicting the timing of an earthquake. To be more specific, the predictions are related to the time left before the next earthquake. To predict this timing, one experiment has been carried out in the laboratory upon a rock in a double direct shear geometry subjected to bi-axial loading as depicted in Fig. 1(a). Two fault gouge layers are sheared simultaneously while subjected to a constant normal load and a prescribed shear stress. The acoustic data is recorded by a piezoceramic (PZT) sensor (as displayed in Fig. 1(a)). The laboratory faults fail in repetitive cycles of stick and slip that is meant to mimic the cycle of loading and failure on tectonic faults which is depicted in Fig. 1(b). However, the timing of an earthquake is based on a measure of fault strength. When a laboratory earthquake occurs, this stress drops unambiguously.



(a) Experiment setup

(b) Results obtained from shear stress

Fig. 1. The laboratory experimental setup along with the results obtained from shear stress [1].

Many methods have been proposed in order to predict the earthquake [2–5]. However, all these methods use traditional machine learning algorithms. Recently, due to the massive amount of data and highly powerful graphical processing units (GPUs), deep learning (especially deep neural network (DNN)) have obtained remarkable performance than that of traditional machine learning algorithms. In DNN, a long short-term memory (LSTM) which was proposed by Hochreiter et al. [6] is used for the prediction task. After that, a simple version of LSTM called gated recurrent unit (GRU) is proposed by Cho et al. [7]. We perform the earthquake prediction over these models and found that GRU is favoured over LSTM as the former is comparatively computationally cheaper and has fewer parameters to train. In this paper, we introduce DNN based models based on a different combination of GRUs to predict the time to the next earthquake.

The main contributions in this manuscript are as follows:

- We propose different deep neural network models based on GRU which can predict the time to next earthquake precisely.
- We compare the different combination of DNN based models and analyze them with different training strategies.

2. Related Work

Many machine learning algorithms have been proposed in the literature to predict the earthquake. Rouet-Leduc et al. [2] proposed a model which can predict quasi-periodic laboratory earthquakes with continuous acoustic data by using a Random Forest method. These authors have further extended their work and proposed a new model in [3] which infer the fault zone frictional characteristics and predict the state of stress by using a gradient boosted tree approach. Recently, authors in [4] conducted one experiment upon slow earthquakes in the Cascadia subduction zone by employing a random forest method [8]. Hulbert et al. [5] conduct another experiment to study the similarity between slow and fast earthquakes using a gradient boosted tree algorithm [9].

Recently, deep learning (especially recurrent neural network (RNN)) have achieved a remarkable performance in the prediction tasks. The RNN model is further modified in terms of long short-term memory (LSTM) [6] as an improvement to the RNN architecture as it solved the problem of exploding/vanishing gradient and was able to remember long sequences which RNN can not able to accomplish. Recently, Shi et al. [10] propose a combination of convolution and LSTM based end-to-end trainable model for the precipitation nowcasting problem and proved that their model outperforms the simple LSTM model. The LSTM model was further simplified in terms of the gated recurrent unit (GRU) [7] where a GRU cell is similar to an LSTM cell except that the GRU cell does not require an output gate. Ballas et al. [11] propose a model by combining the GRU and convolution layers for learning the video representation. Similarly, Zhang et al. [12] introduce convolution-GRU combined model to detect the speech on twitter. In this work, we use different deep learning models to predict the time left to an earthquake. From the best of our knowledge, we can say that it is the first work in which deep learning is used to predict an occurrence of the earthquake.

3. Methodology

The network architecture of the proposed models is display in Fig. 2. Here, five different architecture designs are displayed. The first model utilizes five fully dense layers. While the second and third model consists of five LSTM and GRU layers, respectively followed by one dense layer. The last two models are the two variants with a combination of convolution and GRU layers. Here, the first variant is based on two GRUs followed by one convolution layer while the second variant is based on one convolution layer followed by two GRU layers. In both these variants (see the forth and fifth model in Fig. 2), combination layers are followed by two dense layers in order to map the final output. The kernel size is set to 1 with stride value of 1 in all 1-D convolution layers. In all these models, the network design of LSTM and GRU is same as described in papers [6, 7].

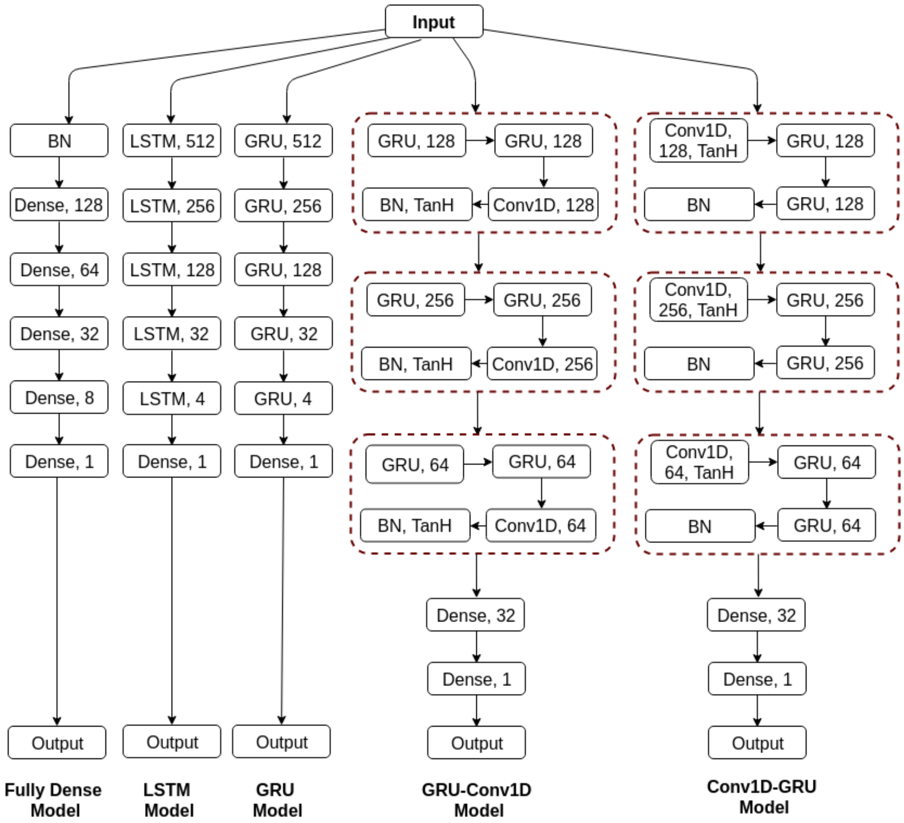


Fig. 2. The network architecture of the proposed models. Here, BN represents the batch normalization layer. The number in each layer block indicates the number of feature maps.

Initially, the acoustic data was fed as input into the neural network models. To increase the forecasting capability on the testing data, Gaussian noise of mean 0 and width 0.5 is added to each chunk of 150,000 data points, then high-frequency noise present in the chunk is removed with the use of the Wavelet Transform using a 4th order Daubechie wavelet [13]. Out of the 17 seismic cycles present in the training dataset, the laboratory earthquakes which had similar distributions to the ones present in the test set were considered. Therefore only 1st-3rd, 5th, 10th-12th and 14th-15th seismic cycles are considered out of the 17 present in the training set. Once a chunk has been cleaned of high-frequency noise, seventeen acoustic features are derived from the acoustic data, which are then fed into the models. The model takes these seventeen features as input and predicts the time left to the earthquake as output.

4. Result Analysis

The acoustic dataset used for the experiments is publicly available at kaggle [1]. In training dataset, there is 629,145,480 number of data points. The testing dataset consists of 2624 number of segments where each segment has 150,000 data points. Each data point represents one acoustic data. This acoustic dataset is pre-processed as discussed in section 3 and then the processed dataset is fed into the proposed models. All the proposed models have a batch size of 64. They are trained using the L 1 loss function. Here, L 2 regularization of 0.01 is applied to prevent the model from overfitting. The Adam optimizer [14] is used with the learning rate left at its default value of 0.001. The total number of epochs defined for training is 1,000. Here, the mean absolute error (MAE) is used as an evaluation metric for validating the performance of the proposed models. The predicted outputs obtained from the proposed models are uploaded in kaggle website and then they calculate the MAE value based on the predicted and true values. Lower the MAE value, the prediction is more accurate. All the models are trained on a system with specifications of Nvidia-GeForce 1070 8GB GPU, octa-core CPU with 32 GB RAM. Our implementation is based on Keras with Tensorflow as backend [15].

Table 1 shows the comparison of the different DNN based models in terms of their obtained MAE values. The corresponding number of training parameters are also mentioned in Table 1. In order to observe the effect of the combination of convolution and LSTM layers, we also train two additional models called LSTM-Conv1D and Conv1D-LSTM models. In both of these models, we replace the GRU unit (as depicted in the last two models in Fig. 2) with the LSTM model. The number of the feature maps and other hyper-parameters is kept same as GRU-Conv1D and Conv1D-GRU models. From Table 1, one can observe that the fully dense model have fewer trainable parameters but the MAE value is higher than that of other models. Here, the GRU-Conv1D model obtains best MAE value with less number of training parameters than other models (except Fully Dense model). One can also found from Table 1 that the GRU model has better MAE value with less number of training parameters than LSTM model.

Table 1. The comparison of the different DNN based models in terms of their MAE value.

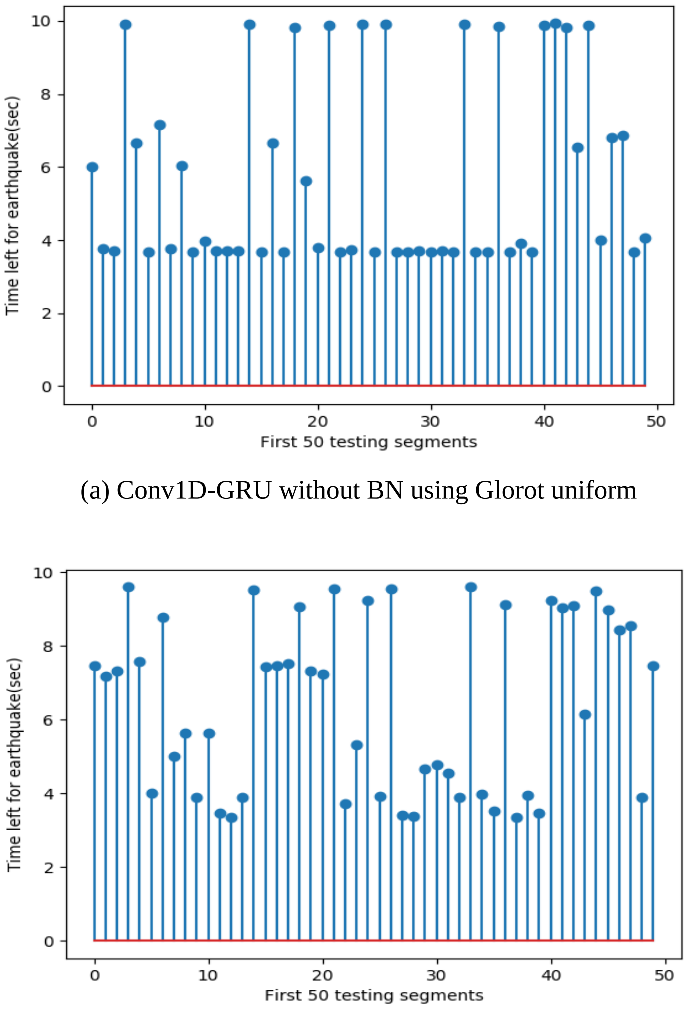
Models	**Numer of Parameters**	**MAE value**
Fully Dense	43,637	3.43682
LSTM	1,313,549	3.52085
GRU	1,751,397	3.39659
LSTM-Conv1D	1,359,617	3.39670
GRU-Conv1D	1,050,497	2.64390
Conv1D-LSTM	1,500,545	3.39670
Conv1D-GRU	1,154,689	3.39662

Table 2. The comaprison of combination of GRU and convolution layer based models for different initialization stretagies and without BN layer.

Models	**Weight Initialization strategies**			**Without BN layer [16]**
	Glorot Uniform [17]	**He Uniform [18]**	**He Normal [18]**	
GRU-Conv1D	2.64390	3.39679	3.82095	3.39673
Conv1D-GRU	3.39662	3.39673	3.39660	2.64607

From Table 1, one can notice that the GRU combination model outperforms LSTM combination models in terms of MAE measure with less number of training parameters. Hence, in Table 2, we compare only two GRU combination variants on three weight initialization strategies named Glorot uniform [17], He uniform [18] and He normal [18]. One can notice from the Table 2 that the Glorot uniform weight initialization performs better in case of GRU-Conv1D model, while for Conv1D-GRU model, all three weight initialization strategies obtain similar performance. This proves that Glorot weight initialization is best weight initialization strategy in GRU-Conv1D model to predict the earthquake. To learn the effect of BN layer, we also train both the models (i.e, GRU-Conv1D and Conv1D-GRU) without using BN layer using Glorot weight initialization. Here, one can observe that the GRU-Conv1D model without BN layer degrades the performance of prediction. Also, the Conv1D-GRU model without BN layer obtains similar performance with that of GRU-Conv1D model. This happens because the BN layer performs better only when it follows by any activation function.

From Table 2, we found that the GRU-Conv1D with BN layer and Conv1D-GRU without BN layer obtains better MAE measures. Hence, the comparison of Conv1D-GRU without BN layer and GRU-Conv1D with BN layer models is depicted in Fig. 3. Here, both these models are compared in terms of time to predict the next earthquake for the first 50 samples of testing dataset.



(a) Conv1D-GRU without BN using Glorot uniform

(b) GRU-Conv1D with BN using Glorot uniform

Fig. 3. The performance comparison of Conv1D-GRU and GRU-Conv1D models in terms of time to failure value of first 50 testing samples.

5. Conclusion

In this paper, we have prepared different deep learning based models to predict time to earthquake based on the acoustic data obtained from laboratory setup. To the best of our knowledge, this is the first work of deep learning to predict the time left to the earthquake. The acoustic data is pre-processed and fed into different deep neural network based models. These models predict the time left for the earthquake. From different experiments, we conclude that the GRU combinations have better performance than that of LSTM combinations with the fewer number of training parameters. We train the proposed model on different weight initialization strategies and found that the Glorot uniform weight initialization performs better than that of others. In order to learn the effect of BN layer, we further train the proposed models without BN layer and observe that the BN layer followed by activation function helps to improve the performance of prediction. Here, we also found that the Conv1D-GRU model without BN layer has similar performance with that of GRU-Conv1D model with BN layer.

References

1. “Lanl earthquake prediction,” https://www.kaggle.com/c/LANL- Earthquake-Prediction.
2. B. Rouet-Leduc, C. Hulbert, N. Lubbers, K. Barros, C. J. Humphreys, and P. A. Johnson, “Machine learning predicts laboratory earthquakes,” Geophysical Research Letters, vol. 44, no. 18, pp. 9276–9282, 2017.
3. B. Rouet-Leduc, C. Hulbert, D. C. Bolton, C. X. Ren, J. Riviere, C. Marone, R. A. Guyer, and P. A. Johnson, “Estimating fault friction from seismic signals in the laboratory,” Geophysical Research Letters, vol. 45, no. 3, pp. 1321–1329, 2018.
4. B. Rouet-Leduc, C. Hulbert, and P. A. Johnson, “Continuous chatter of the cascadia subduction zone revealed by machine learning,” Nature Geoscience, vol. 12, no. 1, p. 75, 2019.
5. C. Hulbert, B. Rouet-Leduc, P. A. Johnson, C. X. Ren, J. Rivière, D. C. Bolton, and C. Marone, “Similarity of fast and slow earthquakes illuminated by machine learning,” Nature Geoscience, vol. 12, no. 1, p. 69, 2019.
6. S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.
7. K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” arXiv preprint arXiv:1406.1078, 2014.
8. L. Breiman, “Random forests,” Machine learning, vol. 45, no. 1, pp. 5–32, 2001.
9. T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. ACM, 2016, pp. 785–794.
10. S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, “Convolutional lstm network: A machine learning approach for precipitation nowcasting,” in Advances in neural information processing systems, 2015, pp. 802–810.
11. N. Ballas, L. Yao, C. Pal, and A. Courville, “Delving deeper into convolutional networks for learning video representations,” arXiv preprint arXiv:1511.06432, 2015.
12. Z. Zhang, D. Robinson, and J. Tepper, “Detecting hate speech on twitter using a convolution-gru based deep neural network,” in European Semantic Web Conference. Springer, 2018, pp. 745–760.
13. A. Graps, “An introduction to wavelets,” IEEE computational science and engineering, vol. 2, no. 2, pp. 50–61, 1995.
14. D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980, 2014.
15. N. Ketkar, “Introduction to keras,” in Deep Learning with Python. Springer, 2017, pp. 97–111.
16. S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” arXiv preprint arXiv:1502.03167, 2015.
17. X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in Proceedings of the thirteenth international conference on artificial intelligence and statistics, 2010, pp. 249–256.
18. K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in Proceedings of the IEEE international conference on computer vision, 2015, pp.1026–1034.