CS230

Deep Learning Application in Japan Earthquake Prediction

Shuojia Fu, Karthik Nataraj, HeeWon Son

Abstract

The accurate prediction of earthquakes can save human life and avoid huge potential costs. Past papers generally predict earthquakes based on the temporal correlation while ignoring the spatial correlation. This paper utilized both multivariate long short-term memory (LSTM) and attention-based networks, which can both incorporate the spatial and temporal relationships among earthquakes. The result shows that the addition of spatial components improves the model performance in the spatial LSTM model, whereas attention models don't yield good results. Also, we implemented a simple LSTM model to determine the influence of extremely large earthquakes (ex:2011 earthquake in Japan).

1 Introduction

Generally happening with minimal forewarning, earthquakes leave people with little time to react, causing enormous death and monetary loss. Lying on the convergence zone of four tectonic plates and containing over ten percent of the world's active volcanoes, Japan experiences on average 1,500 earthquakes with magnitude greater than 4.5 per year and is the hot spot of earthquake over thousands of years[9].Earthquakes causes enormous death and monetary loss in Japan. For instance, on March 11, 2011 a magnitude 9 earthquake occurred in northeastern Japan, destroying countless houses, cars and killing myriad people [9]. The prediction of earthquakes can leave people more time to prepare and evacuate, therefore minimize the potential loss.

Past studies apply models focusing mainly on the temporal correlation among earthquakes,like univariate LSTM models, but neglect the spatial correlation. Within the same fault zone, earthquakes are spatially closely correlated[9]. This paper proposes to utilize both multivariate LSTM and attention-based networks to incorporate the spatial correlation to predict earthquake occurrences in Japan. Japan can be divided into two parts based on fault zone division, and we trained models specifically for the North American plate fault zone.

In terms of input, for the spatial LSTM model the input is a matrix with dimension $M \times L \times N$, where M is the sample size ($M = 564$), L is the length of look-back window ($L = 12$), and N is the number of sub-regions ($N = 4$). Each entry in the matrix is a monthly earthquake frequency in a given location in a given month. The output matrix is of dimension $(M - 12) * N$, with each entry representing the monthly earthquake frequency in a region at the given time. The attention model uses the same L and N as for the spatial LSTM, but for daily data, of which there are 18993 entries, with each entry as a 0/1 binary value to represent whether the earthquake occur or not. Output is a 1×4 vector of predictions. More will be said about the input in the Methods section.

2 Related work

Wang[9] explored the spatial approach for predicting earthquakes in China, achieving 80% accuracy performance just via a LSTM –> feedforward based network. We tried to apply such a model to

CS230: Deep Learning, Spring 2022, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

Japan. Additionally, more sophisticated time series approaches to earthquake prediction have not been seriously explored. We thus explored the use of attention-based models, specifically the use of the Transformer network in [8] adapted to time series. These more general encoder-decoder networks have shown promise in NLP tasks [8], and also been shown to be of value in time series problems, as in [7], [6], [4], and [5], oftentimes outperforming vanilla LSTM/RNN and/or reducing memory requirements, as well as training time. Therefore, in this paper we would like to test whether the attention model can be applied in the earthquake prediction.

3 Dataset and Features

3.1 Raw Data Description

The data was obtained from [1]. We extracted the daily earthquake data in Japan over the past 50 years (1/1/1970-12/31/2021), including time, magnitude, depth, longitude and latitude.

3.2 Data Description for spatial LSTM model

For the spatial LSTM model, we divide the whole North American plate into four equal-area rectangular regions (lower-left, lower-right, upper-left and upper-right), making an array of monthly frequency of earthquake with magnitude greater or equal to 4.5 in each area, and combine the arrays into a $M \times L \times N$ matrix, where M is the sample size ($M = 564$), L is the look-back window ($L = 12$), and N is the number of sub regions ($N = 4$). In addition, we apply a mask value to all the frequency values greater or equal to 80 because only 8 of 564 samples have such extreme values, with the largest one close to 1000. With output as continuous numbers, such anomalies decrease the model performance dramatically. Therefore, we mask such values.

We uses the first 47 years for training, the next 2 years for developing, and the last 2 years for testing. With monthly aggregation, the whole data size shrinks into 612. To have enough training data, we decide to reduce the developing and testing data sizes.We use the most recent four years for developing and testing because we expect them to have the same distribution, whereas the distribution in the precious 47 years could change(ex. 2011 earthquake). In addition, for time series forecasting, utilizing the most recent years for developing and testing can evaluate the effectiveness of the model in predicting present conditions.

3.3 Data Description for LSTM model for extreme case

For the extreme case model, we only apply data after 2011 and used monthly frequency of earthquake with magnitude greater or equal to 4.5. Without spatial components, the input array is $M \times L$ matrix, where M is the sample size ($M = 118$), and L is look-back window ($L = 1$). The first 70 $\%$ of the data were used for training, while each 15 $\%$ of the data were used for the test.

3.4 Data Description for Attention Model

For the attention model, the only difference with the spatial LSTM is that we sample on days and look simply at 0/1 data depending on absence/presence of earthquake with magnitude greater than or equal to 4.5. We divide the data set as 70:15:15 for training, developing and testing sets, respectively. This amounts to 13295 days in the training, and 2849 days each for validation and test.

4 Methods

4.1 Multivariate LSTM Model

Figure 5 in the appendix shows the structure and equations used in a single LSTM memory cell , with i, f , o and c denotes the input, forget, output gates, and cell state. With introduction of forget, input, input modulation, and output gates, LSTM differentiates activation function layers and weighs input differently. Each LSTM recurrent unit maintains a internal cell state that retains the information from the previous unit. The flow of LSTM network resembles the flow of recurrent neural network, with the additional passing of the internal cell state in the hidden layer, which helps LSTM model overcome the vanishing/exploding gradient problems.

The proposed spatial-temporal LSTM model has the structure resembling the multivariate LSTM model.The basic structure has 1 masking layers, 4 LSTM hidden layers each followed by a dropout layer with 0.2 dropout rate and a dense layer with 4 units.The first and the last LSTM hidden layers have 50 neurons, and the rest hidden layers have 100 neurons. We use 12 months look back window, Relu activation function, Adam optimizer and mean squared error loss function.we use the masking layer to reduce the effects of anomaly data; we use the dropout layer to regulate and avoid over-fitting; we use Adam optimizer to increase convergence efficiency in the least time; we use mean squared error loss function since prediction output is continuous.

4.2 LSTM Model for the prediction after the extreme event

Building models with data after extreme event is important because extreme events can change the original distribution pattern. Using LSTM model with 2 hidden layers, 4 nodes, and 1 look-back window, we predicted the frequency of earthquake after the extreme 2011 earthquake. The model did not include any dropout or spatial anaylsis, so the dense layer has one unit.

4.3 Transformer Attention Models

The key theory involved in our attention models is the multi-head attention mechanism. The regular attention function is mapping queries and key-value pairs to an output that is a linear combination of values based on the association between the corresponding key and query. Multi-headed attention is performing this operation in parallel amongst H heads, wherein weight matrices W_i^Q, W_i^K and W_i^V are learned and

$$\text{Multihead}(Q, K, V) = \text{Concat}(h_1, \ldots, h_H)W^O$$
$$h_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V).$$

Instead of a learned attention function the scaled dot-product attention was used. (See [8] for the definition and more information on the specific dimensions of the above matrices.)

We use batch size $B = 64$ and a look-back window of 12 time-steps to predict the next 1×4 vector of earthquake probabilities at the next time-step. Hence our raw input has shape $(64, 12, 4)$. As is typically the case for time series we use the same base input for Q, K and V. We are using either 32-dimensional or 256-dimensional keys (explained below), so in the attention layers inputs are first embedded via linear transformations to be of dimension $(64, 12, 32)$ and $(64, 12, 256)$, respectively.

We used two open-source implementations of the multi-headed attention, both without a decoder layer, but one which has the embedding and the positional encoding [3] and the other which does not [2]. Positional encoding was performed as described in [8] to obtain $(32, 12, 32)$ shaped input to the attention layers. Due to the larger column dimension in this first model we used $H = 4$ and 256-dimensional keys in the second model (without the positional encoding), so the learnt weight matrices there are 4×256 as opposed to 32×32. Finally after the attention matrix is computed W^O is learned and applied to yield $(32, 12, 32)$ and $(32, 12, 4)$, respectively, transformed output.

Besides, there is a simple feed-forward network, dropout, layer normalization, and residual connections between layers happening in each transformer block. We run through this transformer block 4 times in both cases. A nice schematic from the original paper is shown in the appendix, before pooling and applying the sigmoid function to obtain output probabilities.

5 Experiments/Results/Discussion

5.1 Multivariate LSTM Model

We set the batch size as 50, epoch as 200, learning rate as 0.01(as default).Such a combination of hyper parameters yield convergences in loss, and gives us similar and relatively small root mean error square in training and developing sets, which means we are neither over-fitting nor under-fitting.

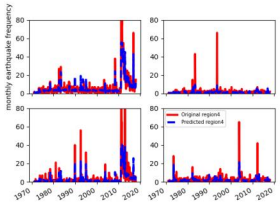
We use root mean square error(RMSE) as the evaluation metric because our output is continuous numbers. For comparison, we do a model with the same structure,but without spatial divisions. Feeding with the same training, developing and testing sets, the RMSEs are 24.849, 9.052 and 4.485, respectively. Compared to the univariate LSTM model with only temporal correlation, the spatial

LSTM yields RMSE that are smaller and more consistent in training, developing and testing sets, validating the importance of spatial correlation.

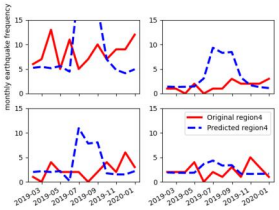
	Whole fault zone	Region 1	Region 2	Region 3	Region 4
Training RMSE	3.995	4.119	3.182	4.248	3.215
Developing RMSE	4.74	7.354	3.764	5.14	1.892
Testing RMSE	5.811	10.847	2.922	2.165	2.044

Figure 1: root mean square error(RMSE) results

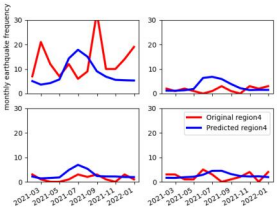
By checking the RMSE scores in each sub region, we found that region 1(upper-left region) has much larger RMSE in the developing and testing sets than the rest regions. With the plots shown below, we found that region 1 has much higher earthquake frequency in the testing and developing sets. Therefore, by using the neighboring earthquake frequency as multivariate input, region 1 has the largest estimation error due to its different distribution from the rest three regions in the testing and training set.



(a) Training results for region 1,2,3,4



(b) Developing results for region1,2,3,4



(c) Testing results for region1,2,3,4

5.2 Extreme Event

For the training, root mean square error(RMSE) was around 15, but for the test, RMSE was around 9.5. The reason to have a larger RMSE in the training than the test is because of the presence of extreme events in 2011 in the training set. The aftershock following the big earthquake in 2011 occurred multiple times and lasted for multiple months, boosting several monthly frequency number in 2011. With lookback window of one month, those high frequencies can hardly be estimated, as shown in figure 3 3.

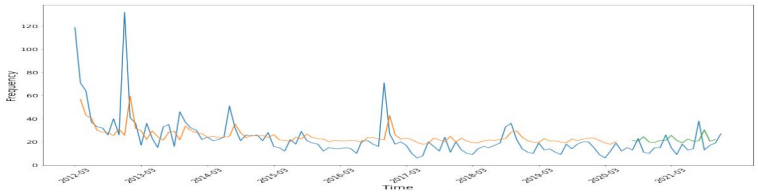


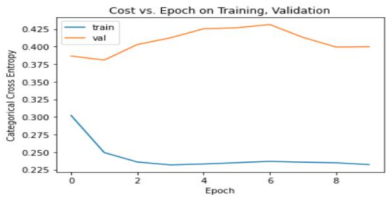
Figure 3: Earthquake Prediction after the Extreme Event

5.3 Transformer Attention Models

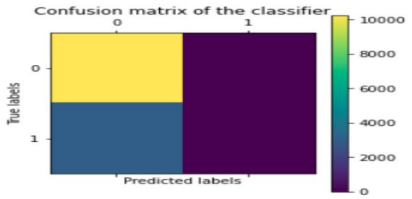
We could not get either attention network to work completely on the spatial data. Though the model without the input pre-processing had more issues. We trained with batch size of 64, as it seemed reasonable for a training set size around 13000. One of these was training loss exploding in early epochs, and corresponding categorical accuracy getting worse (starting from about 40% correct labels decreasing to around 31%.) Looking at the predictions the model was just converging to predicting

labels $(1, 0, 0, 0)$, which is to say an earthquake would happen in the region every day. This was very surprising, as less than 25% of the training data had an earthquake in one of the regions.

The model with pre-processing at least converged on the training set, but very quickly, within 2 epochs (see Figure (a) Cost covergence). It achieved $\approx 77\%$ correct classification accuracy on the training set and $\approx 61\%$ on validation (see Figure (b) Confusion matrix on training data). However we did not move to measuring and improving test set error here still as the model was not outperforming a baseline model that predicts daily, re-occurring non-absence of an earthquake. As there were only earthquakes on about 23% of the days this amounts to a 77% accuracy. We considered that the sparsity was leading to the quick training loss convergence, but even weekly resampling of the data (namely using a dataset with binary labels based on weekly rather than daily occurrence), which had closer to an even split of labels, did not fix this problem. There is also the thought that a more complex network (e.g. one with more parameters) would be needed, though we have not yet obtained significantly better training results increasing H (tried 8 heads, used in [8]), the key dimension (to 512, again from [8]), or hidden layer sizes in the feed-forward network ([8] used 2048, which we did not have resources for. Instead we tried from 32 to 128, in multiples of 32). It is possible that this kind of network does not work well with discrete data and only with continuous series data, as in [2] where on this type of data almost 90% training classification accuracy was achieved.



(a) Cost convergence



(b) Confusion matrix on training data

6 Conclusion/Future Work

The spatial LSTM model yields smaller and more consistent RMSE results for training, developing and testing sets than the traditional LSTM network in our study. Such a result confirms the significance of incorporating the spatial characters into the LSTM network and suggests the possibility of applying such a framework into different countries. In addition, this paper proposes a novel approach to apply attention model to predict the earthquake.However, the model doesn't perform well and we ascribe the reason to be either the requirement for a more complex network or the unfit of such model in earthquake prediction.

For the improvement of spatial LSTM results, we propose to enlarge the data size and test the model performance. Also, we propose to try this model on the other fault zone in Japan. In addition, we propose to divide the fault zones into more sub regions to see whether the error further reduces. Finally, although the overall performance improve, the performance in each sub regions diverge. Further work can investigate how to resolve the divergence of accuracy in each sub regions.

In terms of improving the prediction of these attention networks, there are existing implementations that have appeared in peer-reviewed journals (such as in [4]). These could be explored–the concept is still new with time series data though and use-cases for which it works do vary widely. As for the spatial LSTM it was noticed too late that our architecture might not exactly match that in which has dense networks attached to each hidden state of the LSTM, as opposed to ours for which the dense network is attached only to the final output. Given that such network adds considerable number of parameters to the model it is worth trying to see if it allows us to achieve comparable performance (they achieved about 90% classification accuracy on training data).

7 Contributions

Shuojia worked on spatial LSTM, HeeWon worked on temporal LSTM, and Karthik worked on the attention models. Everyone contributed to their parts of the final report and video.

References

[1] Earthquake catalog. https://earthquake.usgs.gov/earthquakes/search/. Accessed: 2022-05-31.

[2] Timeseries classification with a transformer model. https://keras.io/examples/timeseries/timeseries_transformer_classification/. Accessed: 2022-05-29.

[3] Transformers for timeseries. https://colab.research.google.com/github/charlesollion/dlexperiments/blob/master/7-Transformers-Timeseries/Transformers_for_timeseries.ipynb#scrollTo=S6GCRsxork3M. Accessed: 2022-05-29.

[4] Shengdong Du, Tianrui Li, Yan Yang, and Shi-Jinn Horng. Multivariate time series forecasting via attention-based encoder–decoder framework. *Neurocomput.*, 388(C):269–279, may 2020.

[5] Yao Qin, Dongjin Song, Haifeng Cheng, Wei Cheng, Guofei Jiang, and Garrison W. Cottrell. A dual-stage attention-based recurrent neural network for time series prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, page 2627–2633. AAAI Press, 2017.

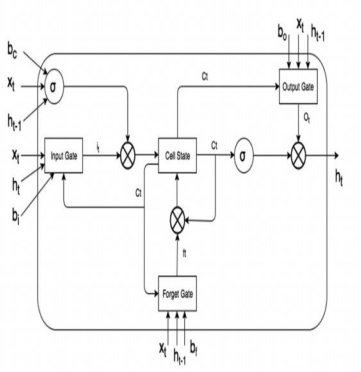
[6] Shun-Yao Shih, Fan-Keng Sun, and Hung-yi Lee. Temporal pattern attention for multivariate time series forecasting. *Mach. Learn.*, 108(8–9):1421–1441, sep 2019.

[7] Huan-Zhi Song, Deepta Rajan, Jayaraman J. Thiagarajan, and Andreas Spanias. Attend and diagnose: Clinical time series analysis using attention models. In *AAAI*, 2018.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.

[9] Qianlong Wang, Yifan Guo, Lixing Yu, and Pan Li. Earthquake prediction based on spatio-temporal data mining: An lstm network approach. *IEEE Transactions on Emerging Topics in Computing*, 8(1):148–158, 2020.

A Appendix 1



$$i_t = \sigma\left(W_{ix}x_t + W_{ih}h_{t-1} + W_{ic}c_{t-1} + b_i\right)$$
$$f_t = \sigma\left(W_{fx}x_t + W_{fh}h_{t-1} + W_{fc}c_{t-1} + b_f\right)$$
$$c_t = f_t \circ c_{t-1} + i_t \circ \phi\left(W_{cx}x_t + W_{ch}h_{t-1} + b_c\right)$$
$$o_t = \sigma\left(W_{ox}x_t + W_{oh}h_{t-1} + W_{oc}c_t + b_o\right)$$
$$h_t = o_t \circ \phi\left(c_t\right).$$

Figure 5: Single LSTM cell and the corresponding equations [9]

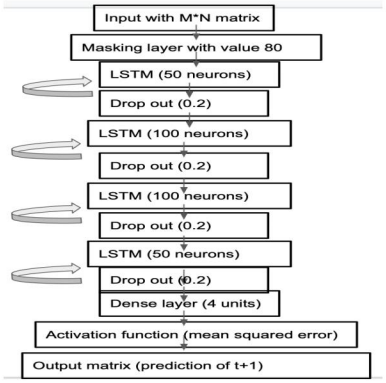


Figure 6: Schematic of Spatial LSTM model

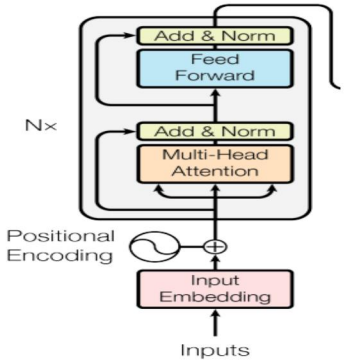


Figure 7: Schematic of attention network