CONCURRENT SEISMIC EVENTS IDENTIFICATION USING DEEP LEARNING

Y. Chen[(1)], M.-H. Yen[(2)], R. Zaccarelli [(3)], D. Bindi [(4)], B. Edwards [(5)], E. Patelli [(6)]

[(1)] *PhD candidate, Institute for Risk and Uncertainty, University of Liverpool, yuchen2@liverpool.ac.uk*
[(2)] *PhD candidate, Institute of Geosciences, University of Potsdam, mingshen@gfz-potsdam.de*
[(3)] *Scientist, GFZ German Research Centre for Geosciences, riccardo.zaccarelli@gfz-potsdam.de*
[(4)] *Senior Scientist, GFZ German Research Centre for Geosciences, dino.bindi@gfz-potsdam.de*
[(5)] *Reader, School of Environmental Science, University of Liverpool, edwardsb@liverpool.ac.uk*
[(6)] *Professor, Department of Civil and Environmental Engineering, University of Strathclyde, epatelli@liverpool.ac.uk*

Abstract

In recent years, a significant increase in the volume and variety of seismological data has been observed. The exponential increase in seismic recordings, facilitated by abundant dense monitoring networks, demands systematic approaches to deal with continuous data-streams, such as procedures for data downloading as well as pre-processing. However, implicit in much of these automated procedures is the reality that often a multi-event segment may arise once a station simultaneously records a near event as well as other events (i.e. teleseismic events or aftershocks), contaminating the main earthquake recording of interest. Significant consequences might occur should this error be unidentified and propagated into subsequent analyses. This paper presents a novel approach to identify multiple seismic events by using Deep Learning techniques in an end-to-end fashion. In particular, we transform the waveforms into time-frequency representations, thereby applying Convolutional Neural Networks (CNN) to automatically pick out the multi-events, without manual feature design. To facilitate uptake and maximise usability, we programmed the proposed methodology in Python so that it could be further integrated to existing pre-processing codes, such as Stream2segment (s2s).

Keywords: seismic events, time-frequency analysis, deep learning, convolutional neural network

1. Introduction

In recent years, there has been a significant increase in dense seismic monitoring networks which provide high quality recordings [1, 2]. The vast volume and variety of such data therefore requires systemic programs to automatically deal with segments from the continuous stream in downloading and pre-processing procedures [3].

A limitation in much of these automated procedures is that there might be some undetected problems with the data [4, 5]. For example, we may find multiple seismic events existing in a predefined window set up to download the desired segment from the continuous stream. Such 'multi-event' records might appear once a station simultaneously records a near event as well as other events (i.e. teleseismic events or aftershocks), thus contaminating the main earthquake recording of interest.

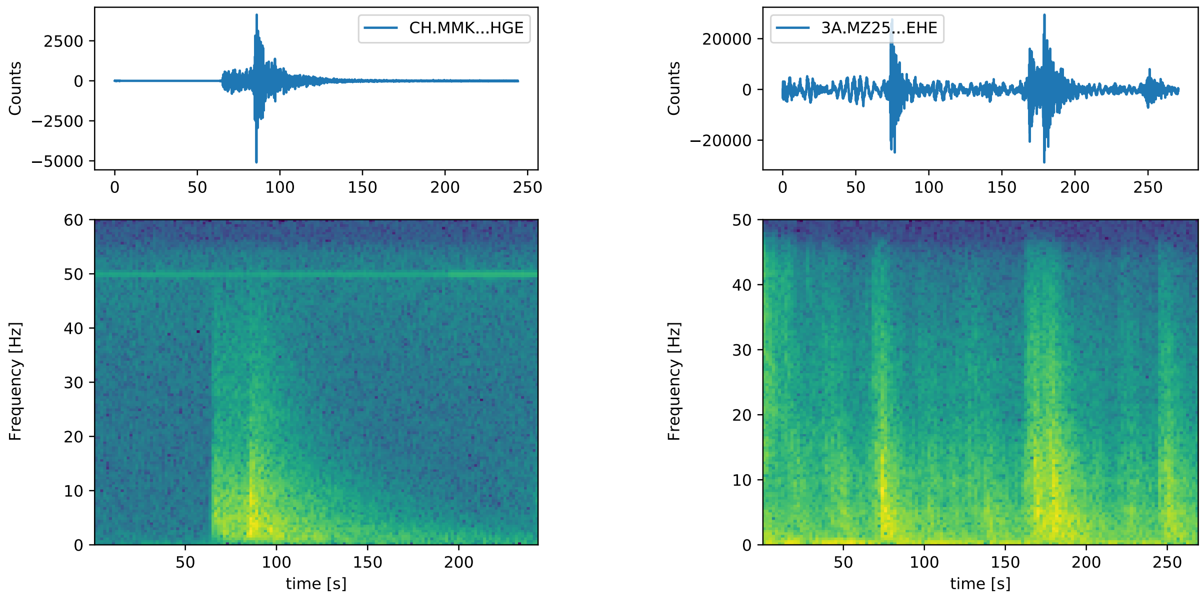
Significant errors and apparent increases in ground motion variability are likely should this error be unidentified during the automatic downloading processes, with the contaminated signal being falsely treated as a clean signal for further spectral analyses. Errors are also likely to be propagated down to subsequent analyses and results, leading to incorrect metadata (magnitude, distance, etc.) and biased models. For instance, it has been found in [6] that significant differences in ground motion variability in several independent geothermal seismicity datasets was likely due to different quality of metadata assigned to each record. In particular, for induced seismicity, low-amplitude signals are susceptible to such disturbances. In some extreme cases, overestimating magnitude due to contaminated data may lead to operation suspension since the ground motion level is heavily monitored and regulated. Usually this is through 'Traffic Light Systems' [7] that impose operational restrictions based on reported earthquake magnitude, which, as noted earlier, may be sensitive to multi-event records.

The challenge, therefore, is that a robust automated procedure that can differentiate unwanted or multiple seismic events is needed. To that end, this paper presents a novel approach using Deep Learning techniques. With Deep Learning, we approach the automated multi-event identification challenge as an image classification problem. The original one-dimension signal is represented as a 2-dimensional pixel-based image through time-frequency analysis. Importantly, compared to traditional signal processing techniques that require hand-crafted feature extraction in the time domain, a CNN model can automatically extract visual features from time-frequency representations.

2. Deep Learning on signal classification

Deep learning allows layered neural networks to directly learn representations from data with multiple level of abstraction. Compared to conventional machine learning approaches that demand domain expertise to elaborately design feature extractors, Deep learning techniques excelled in their ability to process data in their raw form [8, 9].

Recently, significant progress has been made towards signal processing using deep learning techniques [10]. Notably, Convolutional Neural Networks (CNNs) provide new perspectives into processing the otherwise 1D time series data. Examples from diverse fields include audio speech recognition [11], radar signal classification for autonomous vehicles [12], ECG signal-based biometric recognition [13].



(a) a single-event segment (b) a multi-event segment

Fig. 1 –Illustrations of ground motion recordings in different representations. Upper: 1D waveform in the time domain; Bottom: 2D time-frequency spectrogram-like representation by STFT;

2.1 Time frequency representation

The key to the use of CNN on the 1D signal data is to transform it into a two-dimensional representation, i.e. a spectrogram, via time frequency analysis. As a result, the input data then becomes a pixel-based image, from which CNN can be adopted efficiently to extract local visual features for discriminative purposes. Time-frequency representation, by its nature, provides information in both the time and frequency domains, which is especially suited for signals that are characterized with non-stationary features, such as ground motion recordings [14]. The short-time Fourier transform (STFT) is widely adopted to quantify the change of a signal's frequency content over time. A discrete STFT that slides an analysis window over overlapping segments of the signal and calculates the discrete Fourier transform of the windowed data is given by [15, 16]:

$$\mathbf{STFT}\{x[n]\}(m,\omega) \equiv X(m,\omega) = \sum_{n=-\infty}^{\infty} x[n]w[n-m]e^{-j\omega n} \tag{1}$$

where $w[n]$ is a window function (i.e. a Hann window); Varying the window length results in a trade-off between frequency and time resolution. Fig.1 visually shows the ground motion recording in its 1D representation (i.e. waveform) as well as its corresponding 2D representation (i.e. spectrogram-like image by STFT) respectively.

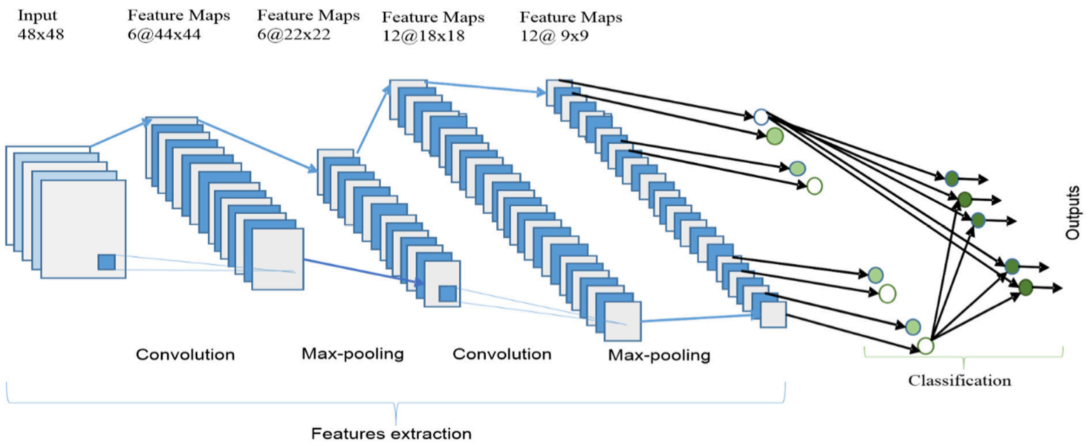


Fig. 2 –A general architecture of CNN, referenced from [17]

2.2 Convolutional neural network

CNN, also known as ConvNet, is focused on detecting and extracting a hierarchy of spatial structures (features) from 2D data arrays - images or spectrograms. The typical architecture of a ConvNet is a stack of several unique layers [8, 18]: a convolutional layer that connects filters with local patches of input image via the convolution operation, a pooling layer that merges semantically similar features into one via subsampling, and a dense layer (fully-connected layer) that leads to the final labels (see Fig.2). By convention, despite the name of convolution, many machine learning libraries indeed implement the operation of cross-correlation, shown as below: Given a two-dimensional input image I with a two-dimensional filter K [9]:

$$S(i,j) = (K*I)(i,j) = \sum_m \sum_n I(i+m, j+n)K(m,n) \tag{2}$$

At the layer level (*a convolutional layer*), on the top of the convolution operation, a non-linearity activation and a bias term are summed. The output, normally called *feature map*, associated with the l-th filter w^l, is obtained from:

$$h_{ij}^l = g((w^l * x)_{ij} + b^l) \tag{3}$$

Where * represents the convolution operation, $g(\cdot)$ is the non-linear activation function and b^l is the bias term. For a binary classification problem as in this case, a binary cross-entropy loss function is used as the goal of loss optimization in the training process.

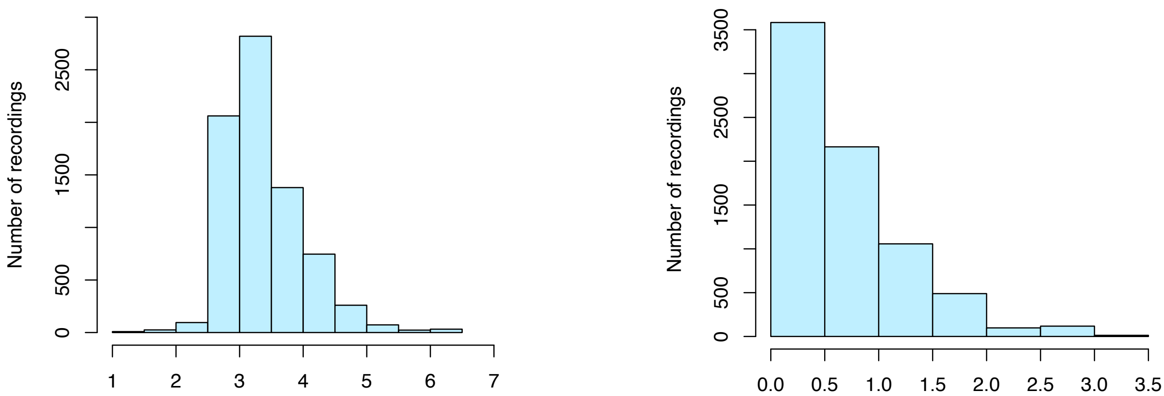
$$L = -\frac{1}{n}\sum_{i=1}^{n} y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i)) \tag{4}$$

Where y_i represents the actual label of the ith instance in the training batch while $p(y)$ stands for the predicted probability $p(y) = \Pr(y=1)$.

3. Classifying double events

3.1 Training dataset

A dataset that contains six thousand manually inspected and labelled seismic waveforms is first established for training purposes. These waveforms, sourced from strong motion or broad-band stations at stable and active tectonic areas across Europe, are downloaded via the tool s2s (see [3]) in its raw form. A variety of magnitudes and distances are considered in the dataset. Fig.3 shows the distribution in terms of magnitudes and hypocentral distances. It should be noted that the scale of degree is used herein as the measure of hypocentral distance.



(a) Distribution of magnitudes (M) (b) Distribution of distance measure (º)

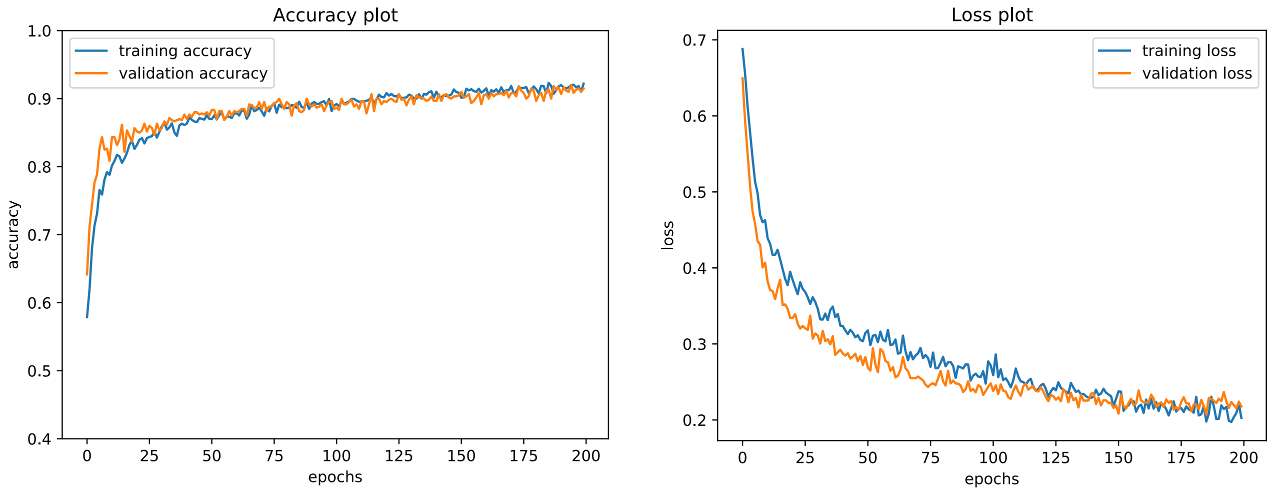
Fig. 3 – Distribution of magnitudes and distance measure in the training dataset

The length of the records' time window is chosen out of a general intention to cover segments with diverse magnitudes and source-to-station distances. To this end, it is routinely set as a 4-minute-long window with one minute preceding the expected P-wave arrival time and 3 minutes after. In this paper, with a predefined four-minute window, a recording with concurrent seismic events is labelled as 1, otherwise labelled as 0 when there is only one normal seismic event. To avoid the bias introduced by the imbalance between classes, this dataset features a 50-50 ratio of both labels (multi events vs single event).

3.2 Results

As a rule, we split our dataset into training, validation, and test sets in a 80:10:10 ratio. Fig.4 shows the progression of performance of our CNN model on the held-out validation data during training, as a function of iterations (epochs), on the scale of accuracy and binary entropy loss (see Eq. (4)) respectively. Each iteration over all the training data is called an epoch while accuracy is defined as the ratio of correctly-labelled instances to the total number of instances. We have shown the learning curves for 200 epochs before the model overfits the training data, by adopting the strategy of early stopping to abort training before the sign of overfitting. A model that overfits would have learnt misleading or irrelevant patterns that don't generalize well from the noises in the data. A model might have started overfitting if the validation loss begins to stagnate while the training loss continues to improve.

As a result, throughout the 200 epochs of training, the training loss curve and validation loss curve align well with each other, moving towards the same direction in which both losses are decreasing at an even slower pace. Correspondently, training accuracy and validation accuracy also align well and progress smoothly in the same direction. They almost converge at the accuracy level of 0.9, suggesting the model has learnt useful mappings from training and does not overfit.



(a) Progression of accuracy during training (b) Progression of binary entropy during training

Fig. 4 – Learning curves during training

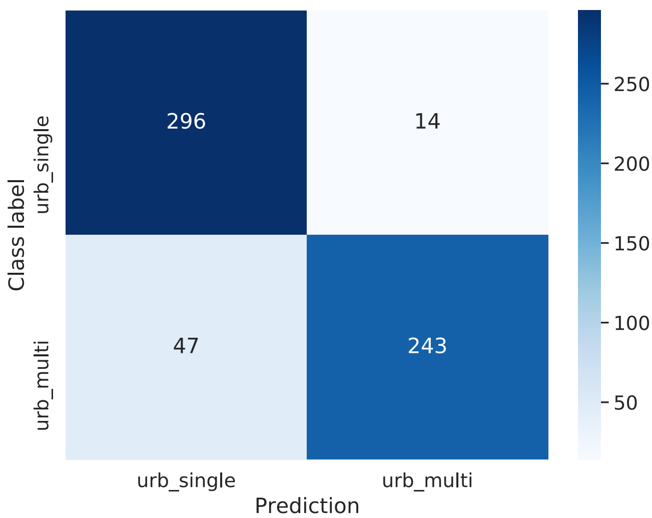


Fig. 5 – The confusion matrix based on test data set

Fig.5 displays the confusion matrix of our CNN model in the held-out test data. Ideally, the confusion matrix would be diagonally dominant, indicating that most predictions have been correctly labelled. Overall, the test accuracy, a unbiased performance measure for a classifier that indicates the generalizing power on unseen data, reaches 0.90, suggesting excellent classifying performance. Upon detailed investigation, we find that three times the number of False Negatives (47) relative to False Positives (14) is observed in this test data (see Fig.5), suggesting it is more likely to falsely think an otherwise double event as a single event, than vice versa. False negatives are commonly called type II errors or overestimation, being interpreted as 'we have missed the hazard' of interest (i.e. double events in this case). Unsurprisingly, this result meets our expectation, since there exists a type of multi-event waveform that have recorded overlapping peaks from two concurrent seismic sources at roughly the same time, which is hard to differentiate with single events even for experienced seismologists.

4. Conclusion

In this paper, we propose a novel approach to detect multiple seismic events in downloading desired segments from the continuous stream. This approach takes advantage of CNN's renowned visual detection abilities on time-frequency representations of seismic recordings. As a result, the multi-event identification challenge becomes an image classification problem. Particularly, through time-frequency analysis (i.e. short-time Fourier transform), we transform the recordings into spectrogram-like representations before feeding into a CNN. Then a CNN model, in a supervised learning scheme, is trained to classify if a recording has multiple events. The CNN model is trained on six thousand labelled waveforms of various magnitudes and distances. Results suggest that our CNN classifier achieves a great performance of 0.91 accuracy on held-out test data.

Due to the fact that an end-to-end CNN model can directly work on data (even on raw data) without hand-crafted feature design, this efficient yet accurate pipeline thereby making a great match with our goal of automatic data downloading and preprocessing. In addition, we programmed the proposed approach in the Python environment (more specifically, Tensorflow) so that it could be further integrated to existing pre-processing programs, such as Stream2segment, to streamline workflow.

Finally, it is acknowledged that our strategy of implementing the CNN on time frequency representation proves to be an efficient and pragmatic solution. But many times we are still facing "bad" data problems in which raw signals might be noisy or incomplete. In theory it will be an even harder challenge for a classifier to tell a single/multi-event-segment in a noisy environment as noises might confuse the identification of seismic events, especially when dealing with low-amplitude waveforms. Furthermore, difficult scenarios where two exactly overlapping signals from two seismic sources are received simultaneously would pose a serious problem to a CNN model. Those lead to our next steps to involve uncertainty measures to give more informed judgement with respect to our multi event identification task. In future works, based on practical concerns, we will investigate the effect of bad data on the performance of the classification task and develop more robust models to work against bad data.

5. Acknowledgements

This research was supported by the EU Horizon 2020 project URBASIS (Project no. 813137).

6. References

[1] Bindi, D., & Kotha, S. R. (2020). Spectral decomposition of the Engineering Strong Motion (ESM) flat file: regional attenuation, source scaling and Arias stress drop. Bulletin of Earthquake Engineering, 18(6), 2581-2606.

[2] Kong, Q., Trugman, D. T., Ross, Z. E., Bianco, M. J., Meade, B. J., & Gerstoft, P. (2019). Machine learning in seismology: Turning data into insights. Seismological Research Letters, 90(1), 3-14.

[3] Zaccarelli, R., Bindi, D., Strollo, A., Quinteros, J., & Cotton, F. (2019). Stream2segment: An open ‐ source tool for downloading, processing, and visualizing massive event ‐ based seismic waveform datasets. Seismological Research Letters, 90(5), 2028-2038.

[4] Obuchowski, J., Madziarz, M., & Zimroz, R. (2015). Seismic multiple events–a study on signals' separation. Vibroengineering PROCEDIA, 6, 212-216.

[5] Maranò, S., Edwards, B., Ferrari, G., & Fäh, D. (2017). Fitting earthquake spectra: Colored noise and incomplete data. Bulletin of the Seismological Society of America, 107(1), 276-291.

[6] Douglas, J., Edwards, B., Convertito, V., Sharma, N., Tramelli, A., Kraaijpoel, D., ... & Troise, C. (2013). Predicting ground motion from induced earthquakes in geothermal areas. Bulletin of the Seismological Society of America, 103(3), 1875-1897.

[7] Bommer, J. J., Oates, S., Cepeda, J. M., Lindholm, C., Bird, J., Torres, R., ... & Rivas, J. (2006). Control of hazard due to seismicity induced by a hot fractured rock geothermal project. Engineering Geology, 83(4), 287-306.

[8] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436-444.

[9] Goodfellow, Ian, Bengio, Yoshua, & Courville, Aaron. 2016. Deep Learning. MIT Press. http://www.deeplearningbook.org.

[10] Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., & Muller, P. A. (2019). Deep learning for time series classification: a review. Data Mining and Knowledge Discovery, 33(4), 917-963.

[11] Deng, L., & Li, X. (2013). Machine learning paradigms for speech recognition: An overview. IEEE Transactions on Audio, Speech, and Language Processing, 21(5), 1060-1089.

[12] Angelov, A., Robertson, A., Murray-Smith, R., & Fioranelli, F. (2018). Practical classification of different moving targets using automotive radar and deep neural networks. IET Radar, Sonar & Navigation, 12(10), 1082-1089.

[13] Labati, R. D., Muñoz, E., Piuri, V., Sassi, R., & Scotti, F. (2019). Deep-ECG: Convolutional neural networks for ECG biometric recognition. Pattern Recognition Letters, 126, 78-85.

[14] Liu, S. C. (1970). Evolutionary power spectral density of strong-motion earthquakes. Bulletin of the Seismological Society of America, 60(3), 891-900.

[15] Marafioti, A., Perraudin, N., Holighaus, N., & Majdak, P. (2019, May). Adversarial generation of time-frequency features with application in audio synthesis. In International Conference on Machine Learning (pp. 4352-4362). PMLR.

[16] Huzaifah, M. (2017). Comparison of time-frequency representations for environmental sound classification using convolutional neural networks. arXiv preprint arXiv:1706.07156.

[17] Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., ... & Asari, V. K. (2019). A state-of-the-art survey on deep learning theory and architectures. Electronics, 8(3), 292.

[18] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.