An application of neural point processes to geophysical data

Pierre-Alexandre Simon, Radu S. Stoica, Frédéric Sur

An application of neural point processes to geophysical data

Pierre-Alexandre Simon[1], Radu S. Stoica[2], and Frédéric Sur[3]

[1] *Université de Lorraine, CNRS, Inria, IECL & LORIA, F-54000, Nancy, France*
[2] *Université de Lorraine, CNRS, Inria, IECL, F-54000, Nancy, France*
[3] *Université de Lorraine, CNRS, Inria, LORIA, F-54000, Nancy, France*

September 2021

Abstract

The huge amount of temporal data available nowadays in numerous scientific fields requires dedicated analysis and prediction methods. Stochastic temporal point processes are certainly one of the popular approaches available to model time series. While point processes have been successfully applied in many application domains, they need strong assumptions. For instance, the conditional intensity is often supposed to follow a particular parametric function, hence fixing a priori the structure of the events distribution: purely random or independent, clustered or regular. Recent papers investigate the use of models from machine learning dedicated to sequential events analysis, namely recurrent neural networks (RNN). These RNNs are expected to be versatile enough to automatically adapt to the data, without the need for a priori choosing the character of the events distribution. This paper presents a brief introduction to the so-called neural point processes and discusses numerical experiments. In particular, the presented real data application considers seismic data from the Guadeloupe region.

Introduction

Stochastic modelling is maybe one of the most used mathematical tools for geophysical data. Clearly, the main interest of using this modelling framework is to provide understanding and prediction of the phenomena underlined by the considered data. Let us consider in the following that our data is a collection of time occurrences of some geological events. This type of random sequence is called a temporal point process or simply a point process. A first question naturally arising is whether the events are independent, if they are clustered - that is one event may trigger another event, or if the events exhibit regularity or repulsion - an event does not favour that another event occurs immediately after. Once these hypotheses are tested, a particular model can be fitted to the data, and statistical inference can be performed, using the analytical properties and simulation algorithms of the considered process. The key tool of this approach is the conditional intensity. This quantity may be interpreted as the probability that a new event occurs knowing the observed process configuration. Important families of such stochastic processes are fully characterised by the conditional intensity. Among them we mention the Poisson processes, the Hawkes processes and the self-correcting processes. The Poisson processes allow to characterise independent events, the Hawkes process describe rather clustered events, while the self-correcting processes are related to repulsive processes. The simulation algorithms and the inference procedures of these processes is based on the computation of the conditional intensity. For a detailed mathematical presentation of point processes theory and of these particular processes, we recommend the monograph Daley and Vere Jones (2003).

The main drawback of using this approach is that in order to fit a model to the data, the type of the model should be chosen. Even if mixture modelling is chosen, fundamentally this question cannot be avoided.

Very recent works - Mei and Eisner (2017); Omi, Ueda, and Aihara (2019) - propose to use neural point processes to analyse time occurrences from seismic data, financial time series, or textual data. These authors used a Recurrent Neural Network (RNN) to model the conditional intensity of

a point process. The great interest of adopting neural nets based inference if compared with classical statistical analysis is that no choice for the model family should be done.

The aim of this work is to apply and to extend the ideas presented in Omi, Ueda, and Aihara (2019) and to test them for predicting seismic data.

The structure of the paper is the following. Next section introduces fundamental facts regarding point processes and the conditional intensity. After this, the recurrent neural nets and neural point processes are described. The fourth section of the paper is dedicated to the obtained results on simulated and real data. The results on the simulated data show how the neural point process are able to characterise independent, clustered and repulsive point processes. The real data application considers seismic data from the Guadeloupe region. Finally, conclusions and perspective are depicted.

1 Temporal point processes

In the following, we consider temporal point process as an ordered sequence of time occurrences (*i.e.* random events) on the positive real axis. The realisation of the process is represented by $\{t_1, t_2, \ldots, t_n\}$ with $t_i \in \mathbb{R}^+$. Furthermore, the sequence is supposed to be locally finite, that is the number of events in a finite interval is always finite. We are interested in modelling the distribution of these events, after its observation in the interval $[0, T]$.

1.1 Modelling the conditional intensity

Three models of temporal point processes will be presented. Each of these models is able to describe a particular relation among the time events. Hence, the considered processes are the homogeneous Poisson processes (HPP), the self-exciting processes (SEP) also known as Hawkes processes, and the self-correcting processes (SCP), modelling independence, excitation, and inhibition between the occurrence of events, respectively. The conditional intensity function at time t is the rate of event occurrence as a function of the history $\mathcal{H}_{t_n}$ of the n event times preceding t. It is written as:

$$\lambda(t|\mathcal{H}_{t_n}) = \lim_{\Delta t \to 0} \frac{\mathbf{E}(N([t, t+\Delta t))|\mathcal{H}_{t_n})}{\Delta t}$$

where $N([t, t+\Delta t))$ is the number of event occurrence in the time interval $[t, t+\Delta t)$ and $\mathbf{E}(N([t, t+\Delta t))|\mathcal{H}_{t_n})$ denotes the expected number of event occurrence in this interval, knowing the history of event times before time t.

It is possible to describe the density function $f(t|\mathcal{H}_{t_n})$ (resp. the cumulative distribution function $F(t|\mathcal{H}_{t_n})$) of the next arrival event after the time t_n as a function of the conditional intensity function $\lambda(t|\mathcal{H}_{t_n})$, by solving the following integral equation (Daley & Vere Jones, 2008, Sec. 14.3.8 p. 399):

$$\lambda(t|\mathcal{H}_{t_n}) = \frac{f(t|\mathcal{H}_{t_n})}{1 - F(t|\mathcal{H}_{t_n})}$$

This gives

$$f(t|\mathcal{H}_{t_n}) = \lambda(t|\mathcal{H}_{t_n}) e^{-\int_{t_n}^{t} \lambda(s|\mathcal{H}_{t_n}) ds} \tag{1}$$

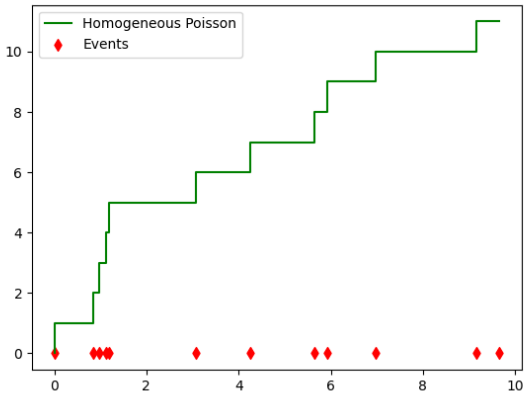
and

$$F(t|\mathcal{H}_{t_n}) = 1 - e^{-\int_{t_n}^{t} \lambda(s|\mathcal{H}_{t_n}) ds}. \tag{2}$$

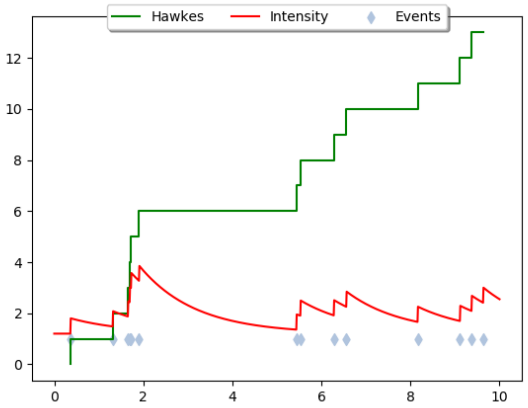
For the HPP process, the conditional intensity is

$$\lambda(t|\mathcal{H}_{t_n}) = \mu_0 \tag{3}$$

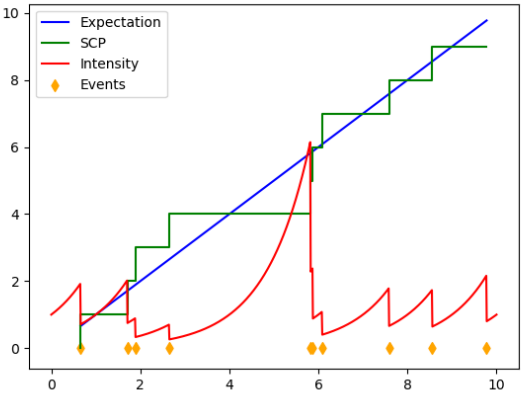
with $\mu_0 \geq 0$ a constant. Note that the case $\mu_0 = \mu_0(t)$ corresponds to independence of events as well. In this paper, we have considered for simplicity only the case where μ_0 is a constant. This case guarantees the same frequency of events in the time interval $[t, t+dt)$ independently of t.



a. Homogeneous Poisson process



b. Self-exciting process



c. Self-correcting process

Figure 1: *a: realisation of an homogeneous Poisson process ($\mu_0 = 1.0$). b: realisation of a self-exciting (Hawkes) process ($\mu_0 = 1.0, \alpha = 0.6, \beta = 0.8$). c: realisation of a self-correcting process ($\rho = 1.0, \alpha = 1.0$).*

In an SEP , the conditional intensity λ is often modelled as the sum of a constant background intensity and a function that combines exponential decays. This last component reflects the excitation involved by past events:

$$\lambda(t|\mathcal{H}_{t_n}) = \mu_0 + \alpha \sum_{i=1}^{n} e^{-\beta(t-t_i)} \tag{4}$$

with $\alpha, \beta > 0$ and $\alpha < \beta$. If $\beta \to \infty$ a SEP tends to a HPP. Otherwise, the intensity encourages the occurrence of a new event t rather close to its ancestors. Hence, this type of process, depending on its parameters, tends to produce clustered configurations of events.

Figure 1 illustrates the realisation of an HP, an SEP, and an SCP. Time events are marked as dots on the horizontal axis. The conditional intensity function $\lambda(t|\mathcal{H}_{t^-})$ is represented in red. In green, the associated counting process is represented. The counting process $N(t)$ is the family of random variables or the stochastic process given by the sum of events in the interval $[0, t)$. Note that the conditional intensity of an SEP is a stochastic quantity.

For an SCP, the conditional intensity tends to inhibit the occurrence of close events. Here, for this purpose, an increasing exponential function was considered

$$\lambda(t|\mathcal{H}_{t_n}) = e^{\rho t - \alpha \sum_{i=1}^{n} \mathbf{1}\{t_i < t\}}. \tag{5}$$

with $\alpha, \rho > 0$ and $\mathbf{1}$ the indicator function.

1.2 Parameter estimation

In this section, a brief and intuitive presentation for the derivation of the likelihood of the considered point processes is given. For a thorough and detailed presentation please refer to (Daley & Vere Jones, 2003, Sec. 7).

The joint probability density of a sequence of n consecutive events may be written as

$$\begin{aligned} f(t_1, t_2, ..., t_n) &= \prod_{i=1}^{n} f(t_i | t_1, t_2, ..., t_{i-1}) \\ &= \prod_{i=1}^{n} f(t_i | \mathcal{H}_{t_{i-1}}) \end{aligned}$$

By plugging (1) and (2) into the preceding equation, the likelihood L of n events occurring in the $[0, T]$ time interval can be expressed as a function of the conditional intensity:

$$\begin{aligned} L &= \left[\prod_{i=1}^{n} f(t_i | \mathcal{H}_{t_{i-1}}) \right] (1 - F(T | \mathcal{H}_{t_n})) \\ &= \left[\prod_{i=1}^{n} \lambda(t_i | \mathcal{H}_{t_{i-1}}) e^{-\int_{t_{i-1}}^{t_i} \lambda(s|H_{s^-}) \mathrm{d}s} \right] e^{-\int_{t_n}^{T} \lambda(s|H_{s^-}) \mathrm{d}s} \\ &= \left[\prod_{i=1}^{n} \lambda(t_i | \mathcal{H}_{t_{i-1}}) \right] e^{-\int_{0}^{T} \lambda(s|H_{s^-}) \mathrm{d}s} \end{aligned}$$

since $t_0 = 0$.

The log-likelihood thus writes:

$$\log(L) = \sum_{i=1}^{n} \log(\lambda(t_i | \mathcal{H}_{t_{i-1}})) - \int_0^T \lambda(s | \mathcal{H}_{s^-}) ds. \tag{6}$$

The above notation $\mathcal{H}_{s^-}$ is denoting the history of the process at time s but excluding s. The parameters of the considered point processes (SEP or SCP) given by (4) or (5) are estimated by maximising the corresponding log-likelihood function. General results and explanations regarding the convexity of the likelihood functions may be found in Daley and Vere Jones (2003).

1.3 Model validation

After the observation of a sequence $t_1, \ldots, t_n$ assumed to be the realisation of a temporal point process, under the hypothesis of a particular chosen model, parameter estimation may be achieved by maximising the corresponding likelihood. The model choice should still be validated. This can be obtained via the results presented in (Daley & Vere Jones, 2003, Thm. 7.4.I and Prop. 7.4.IV). These results guarantee that the transformed point process given by $\Lambda(t_1), \ldots, \Lambda(t_n)$ with

$$\Lambda(t) = \int_0^t \lambda(s | \mathcal{H}_{s^-}) ds \tag{7}$$

is a unit-rate Poisson process. Hence, the inter-arrival times of the transformed process

$$\Big\{ \Lambda(t_2) - \Lambda(t_1),\ \Lambda(t_3) - \Lambda(t_2), \ldots\ ,\ \Lambda(t_n) - \Lambda(t_{n-1}) \Big\}$$

follow an unit-rate exponential distribution. Clearly, plugging into the previous sequence the conditional intensity with the estimated parameters leads to a statistical sample which distribution can be assessed via a Kolmogorov-Smirnov (KS) test. Besides, since Poisson processes give independent inter-arrival times, the significance of the correlation between the transformed inter-arrival times can also be tested with the Ljung-Box (LB) statistical test.

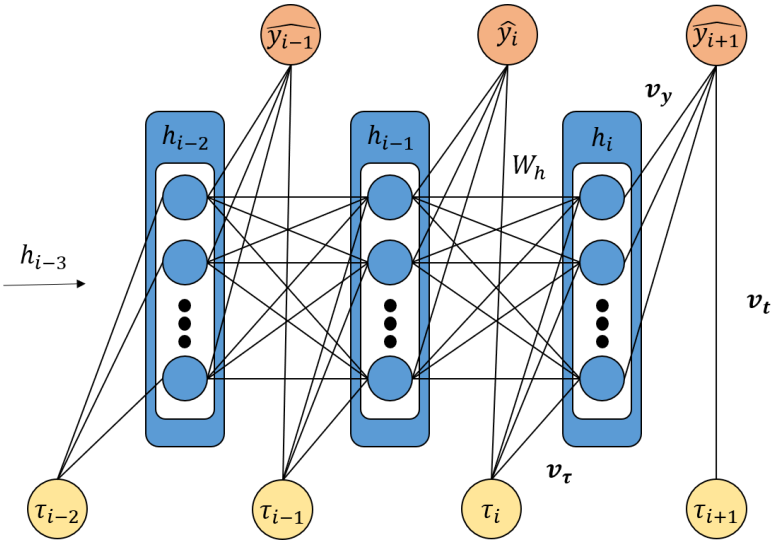


Figure 2: *A recurrent neural network.*

2 Neural point processes

The classical statistical approach for modelling time events using point processes requires the choice of a model. This is not always very convenient, since it requires statistical tests to be done a priori to the parameter estimation. The advantage of using this modelling framework is that it inherits the whole mathematical toolkit in order to rigorously simulate, to predict and to perform a posteriori inference.

At a first glance, the modelling of time events via recurrent neural networks seem to exhibit less mathematical control than the classical statistical approach. Nevertheless, it has the advantage of not depending on the choice of a model, once the structure of the considered neural network is fixed. This section first presents the recurrent neural networks, then their application to modelling of sequential time events based on a construction similar to the conditional intensity of temporal point processes. Finally, the training procedure of such networks is explained.

2.1 Modelling the intensity function with a Recurrent Neural Network

Contrary to the classical models described in Section 1, which are based on a parametric formulation of $\lambda(t|\mathcal{H}_{t^-})$, neural point processes are modelled with a Recurrent Neural Network (RNN). The past of the process taken into account in the intensity function is captured by the so-called internal memory (hidden states h_i) of the RNN at each time-step.

Figure 2 illustrates the RNN architecture and the following notations. Inter-arrival times $\tau_i = t_i - t_{i-1}$ are used for computing the hidden state h_i (which is a vector) at each time-step. The hidden state for the i_{th} time-step is given by:

$$h_i = \sigma_1(W_h h_{i-1} + v_\tau \tau_i + b_h) \tag{8}$$

where σ_1 denotes an activation function (commonly the hyperbolic tangent function), W_h, v_τ, and b_h denote respectively the weight matrix shared between the hidden states, the weight vector for inter-arrival times, and a bias term. The last hidden state is then used for the estimation of the intensity at time t_{i+1}. For instance, an exponential form may be assumed for the intensity, such as:

$$\lambda(t_{i+1}|\mathcal{H}_{t_i}) = \exp(v_y h_i + v_t \tau_{i+1} + b_y) \tag{9}$$

where v_y, v_{tau}, and b_y are respectively a weight vector, a weight parameter, and a bias, h_i a hidden state, and τ_{i+1} the $i+1$-th inter-arrival time.

The output of the network at the i_{th} time-step is eventually given by:

$$\widehat{y_{i+1}} = \log(\lambda(t_{i+1}|\mathcal{H}_{t_i})) = v_y h_i + v_t \tau_{i+1} + b_y \tag{10}$$

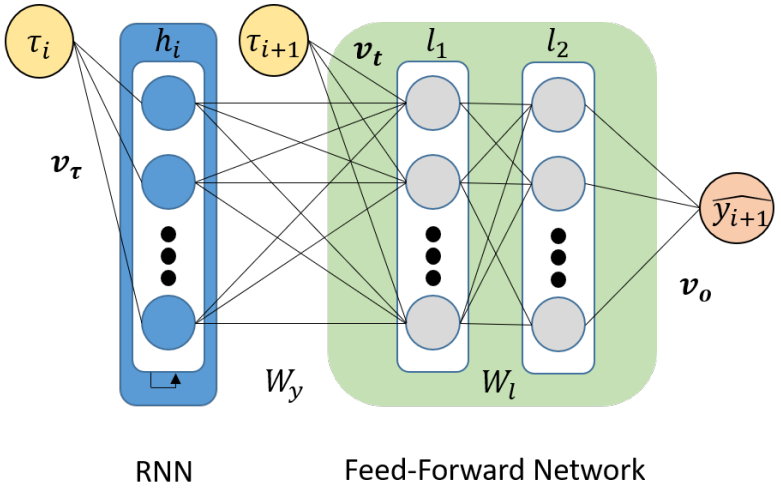


Figure 3: *Architecture of the overall network.*

2.2 Modelling the integral of the intensity function

In RNN, the weights are set during the learning stage (see below) by maximising the log-likelihood function given by Equation (6). However, computing the log-likelihood requires to estimate the integral of the conditional intensity $\lambda(t|\mathcal{H}_{t^-})$ with numerical methods as in Mei and Eisner (2017). Such an approach has a cost in terms of accuracy and computational time. A novel method developed in Omi, Ueda, and Aihara (2019) consists in estimating this integral by giving as an input of a feed-forward neural network the internal state of the RNN modeling the history $\mathcal{H}_t$, together with time t. The reason behind is that a well-suited feed-forward neural network should be able to learn the integral of $\lambda(t|\mathcal{H}_{t^-})$ from samples of t and $\mathcal{H}_t$, since these networks satisfy the universal approximation theorem, see Cybenko (1989); Hornik (1991).

The intensity function is subsequently derived from the integral estimated by the feed-forward network by automatic differentiation. The corresponding neural architecture is described in Figure 3. At time-step $i+1$, the output of the network combining the RNN and feed-forward network is the integral of the conditional intensity function between t_i and t_{i+1} such as:

$$\widehat{y_{i+1}} = \int_{t_i}^{t_{i+1}} \lambda(s|\mathcal{H}_{s^-})ds = \Lambda(t_{i+1}) - \Lambda(t_i)$$

In a similar way as for intensity function, the first part of the network (RNN) is providing a compressed representation of the history $\mathcal{H}_{t_i}$ via the last hidden state h_i:

$$h_i = \sigma_1(W_h h_{i-1} + v_\tau \tau_i + b_h)$$

Then, the second part of the network (the feed-forward network) is modelling the integral of the intensity function taking as input the last hidden state of the RNN and the elapsed time since last event τ_{i+1} such as the output is:

$$\widehat{y_{i+1}} = v_o \sigma_{l_2}(W_l \sigma_{l_1}(W_y h_i + v_t \tau_{i+1} + b_y) + b_l) + b_o$$

where W_y and b_y are respectively the weight matrix and the bias of the connection between the last hidden state h_i and the first feed-forward network layer, and v_t the weight vector of the connection between the i_{th} inter-arrival and the first feed-forward network layer. W_l and b_l are respectively the weight matrix and the bias of the connection between the first and second feed-forward network layer and the vector v_o between the second feed-forward network layer and the output $\widehat{y_{i+1}}$. σ_{l_1} and σ_{l_2} are respectively the activation functions used for the first and the second layer of the feed-forward network.

In all experiments discussed in Section 3, the feed-forward network architecture consists in two hidden layers including each 64 neurons. Its input layer consist in the last hidden state h_i of the RNN and the time elapsed since last event occurrence t. The output layer consists in one neuron, giving a scalar value representing the integral of the intensity function between times t_i and t_{i+1}.

2.3 Neural network training

As in any neural approach, the weights of the network are set according to a data set during the so-called training stage. The sequence of inter-arrival times is split into training and testing sets for the training and evaluation of the network, respectively. In machine learning application, it is important to evaluate models on data-sets which are not involved in training. During the training phase, the log-likelihood function (6) is maximised to derive weights and bias for the neurons of the network, similarly to the standard parametric model. With the notations of Figure 3, the log-likelihood writes:

$$\log(L) = \sum_i \left[\log\left(\frac{\partial \widehat{y_{i+1}}}{\partial t} \right) - \widehat{y_{i+1}} \right]$$

The Back-Propagation Trough Time algorithm is used to obtain the gradient of the log-likelihood function. It consists in unfolding the RNN (the unfolded network is represented on Figure 2) and apply the back-propagation to this unfolded network. The neural network is fed consecutively with the past inter-arrival times considering a moving window of length MW. Considering a fixed-size moving window (hence a limited history) prevents the exploding gradient problem which makes learning to fail. For example, at time step $i+1$, the RNN is fed with the sequence $\{\tau_{i-MW}, \tau_{i-MW+1}, ..., \tau_i\}$. Then, the last hidden state vector h_i provides a compressed representation of this history to the feed-forward network modelling the integral of the intensity function at time $t = \tau_{i+1}$. Finally, the network provides the output value $\widehat{y_{i+1}}$, which represents the integral of the intensity function between time t_i and t_{i+1}.

3 Experiments

3.1 Simulated data

To experiment neural point processes, we generated 100,000 artificial events data from the following point processes, defined by their conditional intensity function λ^*:

- Homogeneous Poisson process ($\mu_0 = 0.2$)
- Self-exciting (Hawkes) process ($\mu_0 = 0.2, \alpha = 0.8, \beta = 1.0$)
- Self-correcting process ($\alpha = 1.0, \rho = 1.0$)

These processes were chosen as they are representative of typical practical situations.

For each process simulated, we computed as well values for the intensity function, its integral and the log-likelihood at each of the arrival times for comparative purposes.

Figure 4 shows the conditional intensity function $\widehat{\lambda}$ derived by neural point process modeling, together with the ground-truth λ. For assessing the quality of fit, we computed different metrics such as the Mean Absolute Error (MAE), Bias, Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) between the intensity evaluated at each new arrival and effective value. The Bias is indicating if we are over(resp. under)-estimating the effective intensity:

$$Bias = \frac{1}{n} \sum_{i=1}^{n} \left(\lambda(t_i|\mathcal{H}_{t_{i-1}}) - \widehat{\lambda(t_i|\mathcal{H}_{t_{i-1}})} \right)$$

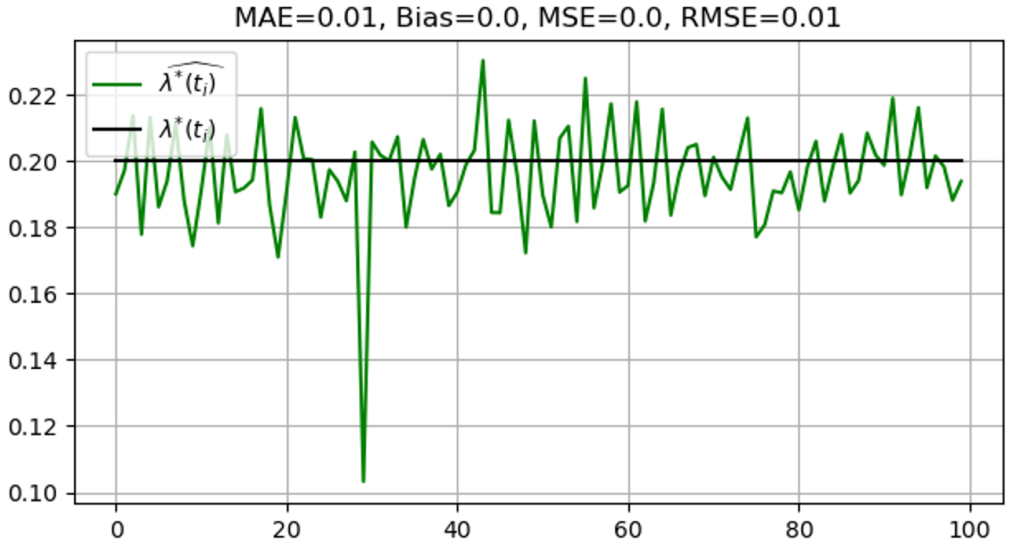


Figure 4: *Neural network estimation of intensity function (green) against intensity function of Poisson process generating artificial data (black).*

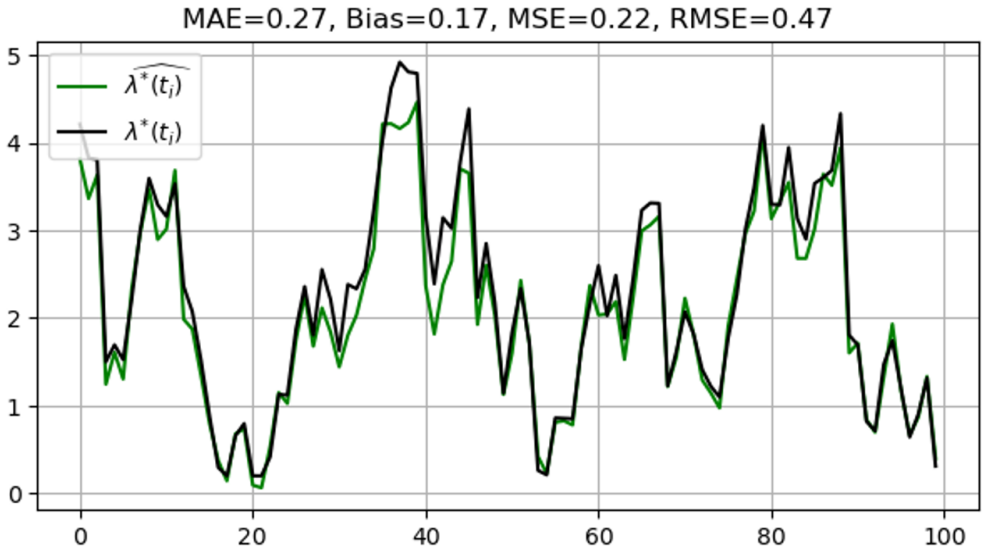


Figure 5: *Neural network estimation of intensity function (green) against intensity function of Hawkes process generating artificial data (black).*

The RMSE penalises heavily significant differences on the intensity function evaluation (by squaring them), which is the main challenge for the Hawkes model when clusters of points form.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\lambda(t_i|\mathcal{H}_{t_{i-1}}) - \widehat{\lambda(t_i|\mathcal{H}_{t_{i-1}})}\right)^2}$$

The RMSE between the estimation and the real intensity is lower than 10% of the mean intensity (equal to 5), showing that neural point process fits well the challenging Hawkes process.

We also computed the Kolmogorov-Smirnov p-value and Kullback-Leibler and Jensen-Shanon divergences for verifying the adequacy of the residual with the unit exponential distribution (as explained in Section 1.3), see Table 1. In the three typical cases, the p-value of Kolmogorov-Smirnov test are

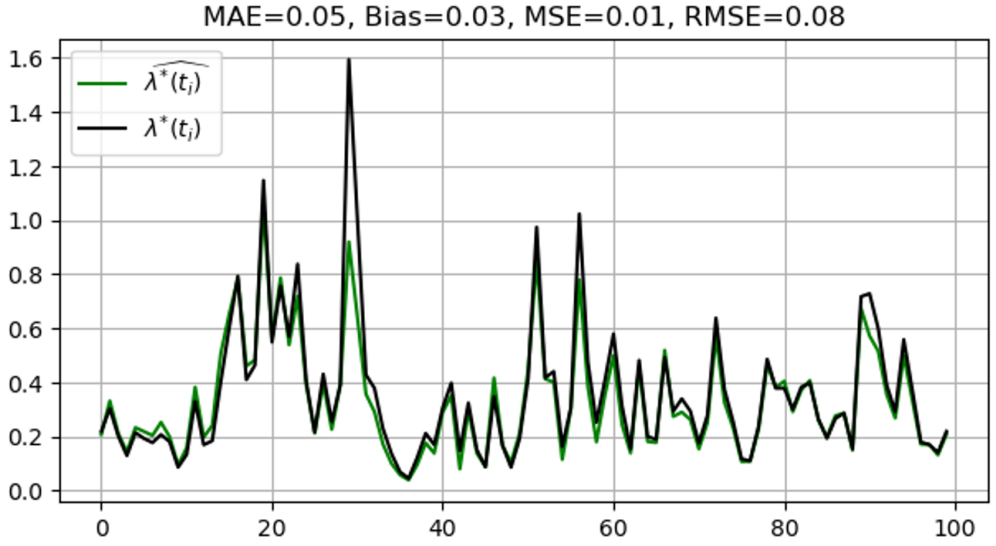


Figure 6: *Neural network estimation of intensity function (green) against intensity function of self-correcting process generating artificial data (black).*

higher than the risk confidence level $\alpha = 0.05$, which indicates that the null-hypothesis (adequacy of the empirical distribution with the unit-rate exponential distribution) cannot be rejected. The small values for KL and JS divergences are confirming the goodness-of-fit of the neural point process for each kind of simulated data. In a similar way, the p-value of Ljung-Box statistical test are higher than the risk confidence level for each case, indicating that the null-hypothesis (inter-arrival are not auto-correlated) cannot be rejected.

Metrics	Poisson H	Self-correcting	Hawkes
RMSE $\lambda^*(t)$	0.01	0.08	0.47
Bias $\lambda^*(t)$	0.0	0.03	0.17
KS p-value	0.48	0.12	0.38
KL divergence	-0.001	0.018	$-1.1.10^{-3}$
JS divergence	$-1.49.10^{-5}$	$-4.3.10^{-4}$	$-5.07.10^{-6}$
LB p-value	0.32	0.77	0.12

Table 1: *Evaluation metrics for each neural point process models*

3.2 Real data

We also proposed to experiment neural point processes on a real data-set gathering occurrence times for 2,264 earthquakes in Guadeloupe between 2004 and 2005 collected by the RING team from Geo-Ressources (see also Ben Allal, Lejay, and Stoica (2020)), illustrated in Figure 7. This data-set is divided into a training set (incl. 1,811 arrival times) and a validation set (incl. 453 arrival times). As Hawkes model is popular for modelling earthquake events, we proposed to compare the neural network model with a specific parametric Hawkes model whose parameters (cf Equation (4)) were estimated with the maximum likelihood method on the training set.

After training the network (with a moving window of 10 elements, as mentioned in Section 2.3), the intensity function (shown in Figure 8), its integral and the log-likelihood functions are computed at each occurrence time of the testing set for both the neural point process model (green) and the parametric Hawkes model (salmon). At a first glance, we observe in Figure 8 that the intensity is capturing well the excitation phenomenon both for parametric and neural point process models. We

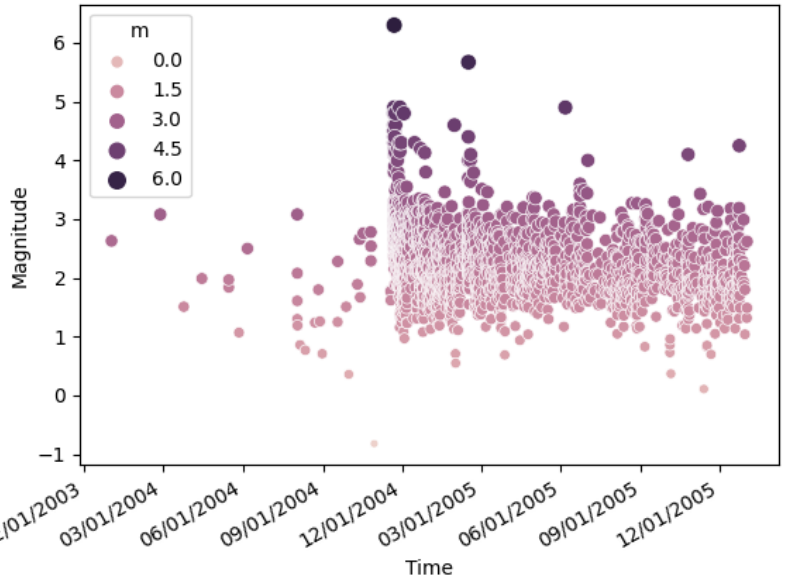


Figure 7: *Earthquake occurrence times and magnitudes.*

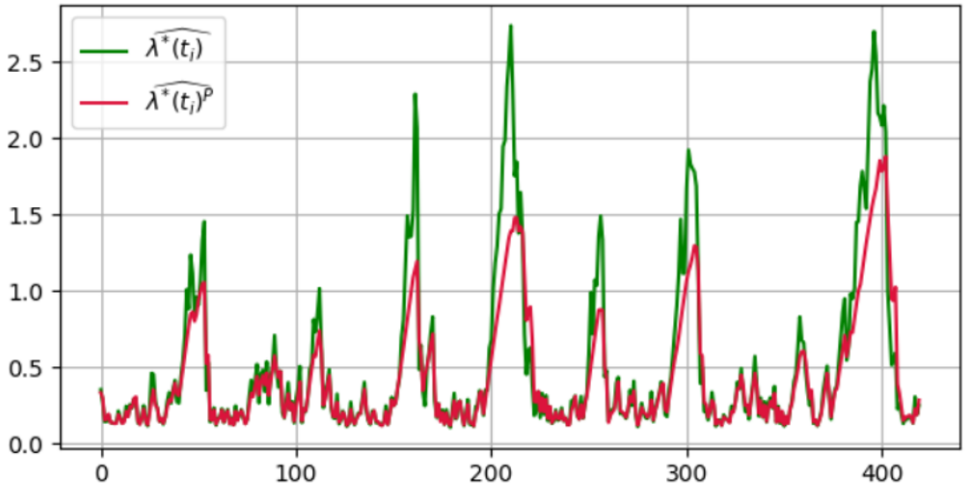


Figure 8: *Derived intensity functions from testing set. The neural network estimation is in green and the standard parametric estimation is in salmon.*

observe that the neural point process over-estimates the intensity peaks due to excitation effects. This is due to the few amount of data at disposal for training, when a larger amount would provide a smoother estimation as in the parametric model.

From the intensity, we are able to predict the occurrence of the next earthquake given the observation of the last 10 earthquakes using (1). We used the bisection method for neural point process as in Omi, Ueda, and Aihara (2019). It consists in using the property that $\{\forall t > t_i, \Lambda(t) - \Lambda(t_i))\}$ follows a unit-rate exponential distribution over the interval $[t_i, t_{i+1}]$. The predicted next arrival time $\widehat{t_{i+1}}$ is then defined as the median value and is obtained by solving $\Lambda(\widehat{t_{i+1}}) - \Lambda(t_i) = \log(2)$. Practically, we computed the integral of the intensity function from the same inter-arrival input sequence, different value for t_{i+1} until the preceding condition is fulfilled. For parametric Hawkes

processes, we fitted the model with training data and obtained the following parameters estimators ($\widehat{\mu_0} = 0.11$, $\widehat{\alpha} = 0.09$, $\widehat{\beta} = 0.13$). Then, we used the Ogata algorithm to simulate 100 inter-arrival times and take the median of the obtained distribution. We then compared these predicted arrival times with the real earthquake arrival times of the data-set. The different evaluation metric for both model are described in Table 2.

Metrics	Hawkes	Neural point process
MAE	2.78	3.57
Bias	1.28	2.22
MSE	17.09	29.32
RMSE	4.13	5.42

Table 2: *Evaluation metrics for next arrival time.*

We observe that the Mean Absolute Error rises up to approximately 4 hours with a positive bias indicating that both models are under-estimating the next earthquake arrival time of 4 hours in average. In other words, the next earthquake is occurring in average 4 hours after the time predicted, which is satisfying for safety purposes. Figure 9 shows a comparison between these results and the real events occurrence times. We can see that short inter-arrival times characterising the excitation phenomenon are well estimated by both models, which is the major challenge when predicting next event occurrence.

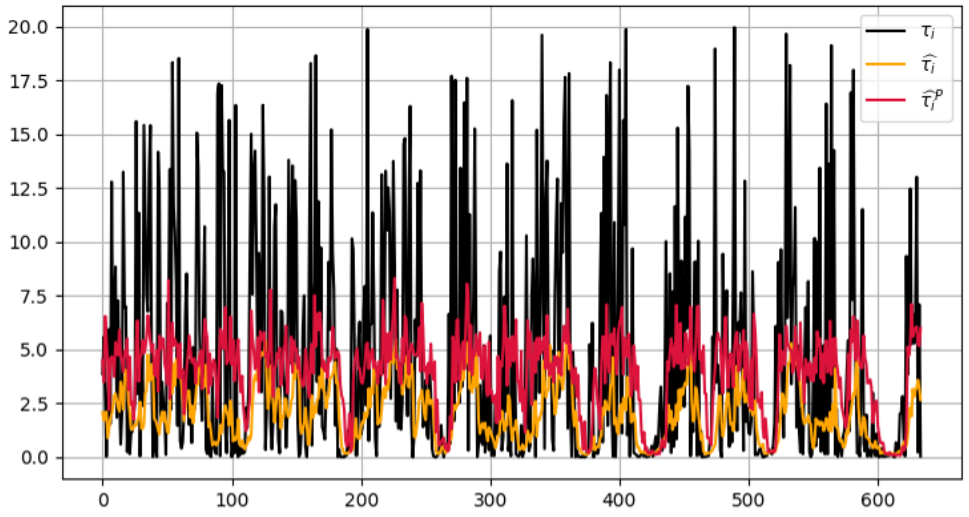


Figure 9: *Next arrival prediction on the testing set. The neural network estimation is in yellow and the standard parametric estimation is in salmon. Actual next arrival time is in black.*

It is interesting to note that the neural network performs similarly to the parametric Hawkes model on all metrics. However, while the parametric model requires the full occurrence time history, the neural network model has only been trained considering the last 10 earthquakes occurrence times (because of the moving window of length 10). Moreover, once the network has been trained, it only needed a few seconds to provide the next arrival prediction sequence where the standard parametric model required 15 minutes. These results are really encouraging since no specific intensity function form is required in the neural point process model. The modest computational resources are also worth mentioning.

Conclusion

Within the neural point process model, we experimented the approach of Omi, Ueda, and Aihara (2019) for modelling the integral of the intensity function of temporal point process with compound neural network. We have shown through numerical experiments on synthetic and real data that neural point processes permit to model in a satisfactory manner three typical situations related to time events: independence, clustering and regular distributions. Compared with traditional point process approaches, no prior assumption on the intensity function is required, the weights of the neural networks automatically fitting the input data in the training stage. Nevertheless, the architecture of involved networks and their hyper parameters as well had to be tuned after several trials and errors.

Acknowledgments

The authors would like to thank the RING team (GeoRessources lab) for kindly providing the seismic data.

References

Ben Allal L, Lejay A, & Stoica R. (2020). Hawkes point processes based inference applied to seismic data analysis. In *Proceedings of the 2020 ring meeting.* (https://hal.archives-ouvertes.fr/hal-02928408)

Cybenko G. (1989). Approximation by superpositions of a sigmoidal function. *Maths. Control Signal Systems*.

Daley DJ, & Vere Jones D. (2003). *An introduction to the theory of point processes, volume 1 : Elementary theory and methods, second edition.* Springer.

Daley DJ, & Vere Jones D. (2008). *An introduction to the theory of point processes, volume 2 : General theory and structure, second edition.* Springer.

Hornik K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks*.

Mei H, & Eisner J. (2017). The neural Hawkes process: A neurally self-modulating multivariate point process. In *31st Conference on Neural Information Processing System.*

Omi T, Ueda N, & Aihara K. (2019). Fully neural network based model for general temporal point processes. In *33rd Conference on Neural Information Processing Systems.*