PhaseLink: A Deep Learning Approach to Seismic Phase Association

Preprint · September 2018

5 authors, including:

Yisong Yue
California Institute of Technology
73 PUBLICATIONS   **1,508** CITATIONS

Men-Andrin Meier
California Institute of Technology
26 PUBLICATIONS   **141** CITATIONS

Egill Hauksson
California Institute of Technology
227 PUBLICATIONS   **6,541** CITATIONS

Some of the authors of this publication are also working on these related projects:

Project   stress drop analysis View project

Project   induced seismicity in California View project

PhaseLink: A Deep Learning Approach to Seismic Phase Association

Zachary E. Ross[1], Yisong Yue[2], Men-Andrin Meier[1], Egill Hauksson[1], Thomas H. Heaton[1]

[1]Seismological Laboratory
California Institute of Technology
Pasadena, CA 91125

[2]Department of Computing and Mathematical Sciences
California Institute of Technology
Pasadena, CA 91125

Abstract

Seismic phase association is a fundamental task in seismology that pertains to linking together phase detections on different sensors that originate from a common earthquake. It is widely employed to detect earthquakes on permanent and temporary seismic networks, and underlies most seismicity catalogs produced around the world. This task can be challenging because the number of sources is unknown, events frequently overlap in time, or can occur simultaneously in different parts of a network. We present PhaseLink, a framework based on recent advances in deep learning for grid-free earthquake phase association. Our approach learns to link phases together that share a common origin, and is trained entirely on tens of millions of synthetic sequences of P- and S-wave arrival times generated using a simple 1D velocity model. Our approach is simple to implement for any tectonic regime, suitable for real-time processing, and can naturally incorporate errors in arrival time picks. Rather than tuning a set of ad hoc hyperparameters to improve performance, PhaseLink can be improved by simply adding examples of problematic cases to the training dataset. We demonstrate the state-of-the-art performance of PhaseLink on a challenging recent sequence from southern California, and synthesized sequences from Japan designed to test the point at which the method fails. These tests show that PhaseLink can precisely associate P- and S-picks to events that are separated by ~ 12 seconds in origin time. This approach is expected to improve the resolution of seismicity catalogs, add stability to real-time seismic monitoring, and streamline automated processing of large seismic datasets.

1 Introduction

When an earthquake is detected on different stations of a seismic network, it is often desirable to link the observed seismic phases to the earthquake that caused them. Historically, this task was performed by expert seismic analysts who would visually examine the data from different stations, identify seismic phases, and group them together (cf. Figure 1). As the modern digital era began, seismic networks started to accumulate data in real-time, and it became necessary to develop computer algorithms to automatically process the data.

With the development of the landmark STA/LTA algorithm in seismology [1, 2], it became possible to detect earthquakes automatically for the first time. This simple method uses the ratio of two moving averages to identify impulsive transient signals and has become the de facto standard for earthquake detection around the world. One major shortcoming of the method is that it will not only identify earthquakes when present, but also any other types of impulsive transient signals that
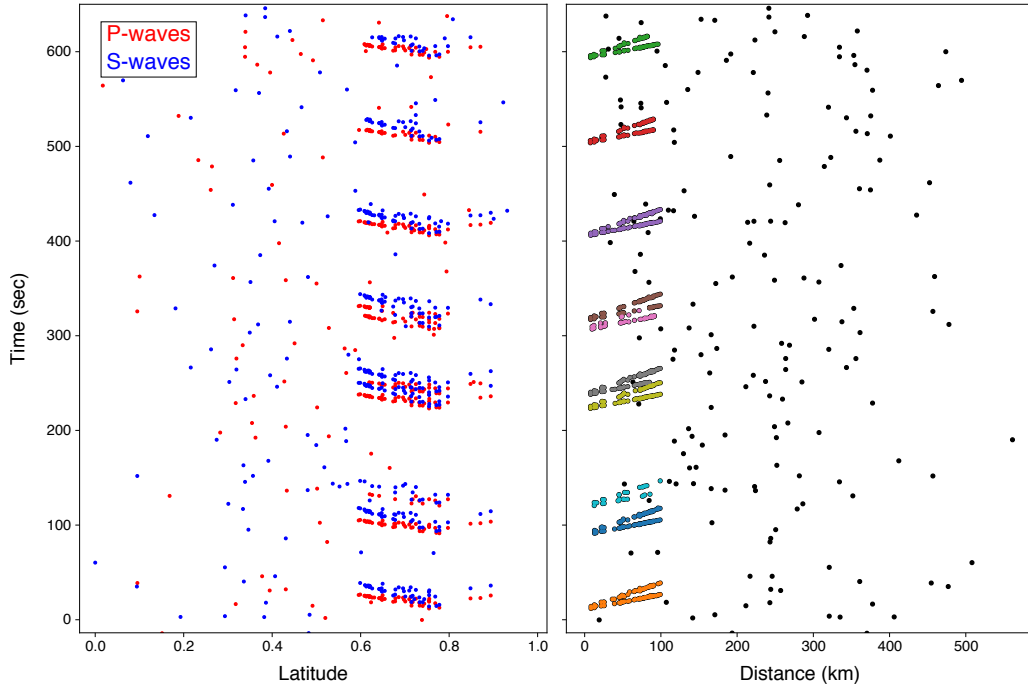
Figure 1: Cartoon example of a phase association scenario. Left panel shows the discrete set of picks for the entire network. The number of events is unknown. Right panel shows the output after association and location. Picks colored black are not linked to an event, while colored picks share a common origin.

seismometers record. This led to the development of the first phase association algorithms, which examine combinations of triggers on different stations to see whether any set have arrival time patterns consistent with those of earthquakes [12]. The association process therefore evolved from one of simply grouping seismic phases together, to being ultimately responsible for deciding whether an earthquake occurred.

To date, algorithms for phase association all operate using the same fundamental principle. The region of interest is gridded and for each node therein, tentative phase detections within the network are examined to see whether some subset back-projects to a coherent origin. This means that a grid search must be conducted continuously for all new picks that are made. Typically grid associators require extensive tuning of a large number of sensitive hyperparameters, and have numerous ad hoc rules to stabilize potential problems that can arise. Over the years, they have become increasingly sophisticated, with modern variants incorporating Bayesian estimates of pick uncertainties [17], or multi-scale detection capabilities.

Today, seismologists strive to identify increasingly smaller events that are often at or below the noise level. Resolving this level of detail requires not only increasing phase detection sensitivity, but dealing with the dramatically larger volume of information to be processed in a reliable and rational manner. In particular, since smaller events occur ever more frequently, and therefore are more closely spaced in time, moving forward requires technology that can easily handle the most complicated scenarios encountered at the present.

In recent years, deep learning has become state of the art in numerous domains of artificial intelligence [15], including natural language processing [28], computer vision [14], and speech recognition [3], and is focused on learning generalized representations of extremely large datasets. Deep learning has been recently used in seismology, and has already shown considerable promise in performing various tasks including similarity-based earthquake detection and localization [18], generalized seismic phase detection [21], phase picking [35], first-motion polarity determination [20], detection of events in

laboratory experiments [33], seismic image sharpening [16], and predicting aftershock spatial patterns [6].

In this paper, we present PhaseLink, which is a deep learning approach for grid-free earthquake phase association. Our approach is built upon Recurrent Neural Networks (RNNs), which are designed to learn temporal and contextual relationships in sequential data. We show how to design a training objective that enables the trained RNN to accurately associate phase detections coming from multiple temporally overlapping earthquakes. Another attractive feature of our approach is that is it trained entirely from synthesized data using simple 1D velocity models.[1] Thus, our approach is easily applicable to any tectonic regime by simply training on the synthesized data from the appropriate model, and can also naturally incorporate errors in arrival time picks. The full source code will be publicly available via the Southern California Earthquake Data Center <scedc.caltech.edu>.

2 Background on Recurrent Neural Networks

Artificial neural networks are systems that can discover complex non-linear relationships between variables. Fundamentally, they successively transform a set of input values through matrix multiplication and non-linear activation functions into one or more output variables of interest [8]. The outputs can be either continuous (regression) or discrete (classification). In supervised learning, the parameters which characterize the non-linear mapping are learned by minimizing the prediction error of the model against the ground truth. The standard type of neural network is today referred to as a fully-connected neural network because each neuron is fully connected to each previous input. Fully-connected networks are excellent at many classification and regression tasks, but have trouble discovering structure in sequential datasets because they lack feedback mechanisms that can enable information to propagate between successive elements of a sequence.

These shortcomings were addressed by the development of the recurrent neural network (RNN) [11]. RNNs allow for information to be passed between successive elements through the use of an internal memory state. This state is dynamically modulated by gates that control what information is retained along the way, and the parameters governing the gates themselves are learned through the training process. The outputs of RNNs are very flexible, and could be a single valued output given an input sequence, or a sequence of outputs. To date, RNNs have been applied to variety of settings, including language translation [28], speech synthesis [30], speech recognition [3], image captioning [34], and many others.

The most commonly employed variant of the RNN is the long short-term memory (LSTM) [10] network. These networks have three gates that control the flow of information, and are useful because they are not so susceptible to training issues related to diminishing propagation of information over large sequences. In recent years, another variant called the gated recurrent unit (GRU) [5] has become popular because it has only two gates instead of three, resulting in fewer parameters and faster training. These types of RNNs are considered state of the art for many problems including speech recognition and language translation.

Over the years, numerous improvements have been made to these basic types of RNN layers, and one such important development was the bidirectional RNN layer [23]. This layer uses two RNNs running in opposite directions so that information from both directions of the sequence is available to make predictions. A common example where this is useful is word prediction, where if a word in the middle of a sentence is missing, it is generally desirable to use the contextual information from the entire sentence to make a prediction, rather than just the words leading up to the missing one.

3 Summary of Datasets Used

In this study, we use synthetically generated seismicity and phase sequences, and real phase data for the 2016 Borrego Springs sequence [19], which occurred in southern California during 2016-06-01 to 2016-06-31 (last accessed August 2018). During this period, 1708 earthquakes were identified by the Southern California Seismic Network (SCSN), and 73,353 phases were picked by seismic analysts. The data are publicly available from the Southern California Earthquake Data Center

---

[1] This paradigm is generically known as "sim-to-real" in the machine learning community [27, 24, 29, 7, 32].

<scedc.caltech.edu>. No waveform data were used. We also used synthetically generated seismicity using the station configuration of the Hi-net seismic network in Japan.

4 PhaseLink Framework

The PhaseLink approach is designed to solve the phase association problem: given a sequence of N picks, determine how many earthquakes (if any) occurred, and which of the N picks belong to each respective earthquake. Fundamentally, one can think of phase association as a (supervised) clustering problem of clustering picks to earthquakes that generated them. In contrast to conventional clustering, there is a specific temporal structure to our prediction task, and also the number of clusters is unknown a priori. For instance, having multiple overlapping earthquakes implies detecting picks coming from different "clusters".

Figure 2 depicts the PhaseLink approach, which can be conceptually described in the following steps:

- We are given an input stream of picks. Each pick has as attributes the location (latitude & longitude) of the station that detected the pick, the time stamp, and phase type (Fig. 2, step 1).

- The input pick stream is processed into a sequence of overlapping fixed-length sequential prediction tasks (Fig. 2, step 2). In particular, the prediction task is whether each pick in the input sequence belongs to the same earthquake that generated the first (root) pick in the sequence, i.e., a sequential binary classification problem. We solve this fixed-length prediction task using RNNs (Section 4.1), and we train the RNNs using synthetic data (Section 4.3).

- The overlapping predictions are then aggregated into a single set of clustering results of picks to earthquakes (Section 4.2; Fig. 2, step 3).

By decomposing the problem in this way, PhaseLink can, in principle, handle any number of overlapping clusters. Conceptually, the reduced prediction task is based around a reference point and classifies a temporal neighborhood of points as belonging to the same cluster as the reference point. A somewhat similar idea was proposed in supervised clustering approaches that utilize must-link and cannot-link constraints [31, 4], although those approaches are more geared towards learning a metric space rather than directly solving the clustering problem. Furthermore, our PhaseLink approach can exploit a natural temporal locality structure to further constrain the prediction task. Another benefit of directly considering the co-clustering prediction problem is that we can tolerate false picks (those that do not belong to any cluster/earthquake). A final benefit of this decomposition is that PhaseLink can utilize off-the-shelf RNN implementations, which leads to significantly reduced system engineering overhead.
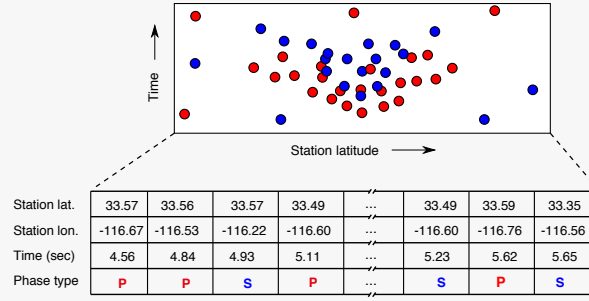
4.1 RNN Architecture

We designed a deep RNN consisting of stacked bidirectional GRU layers (Table 1). The network takes as input fixed-length sequences of picks, and outputs a sequence of identical length (Fig. 2, step 2). The output sequence is binary valued, with a value of 1 indicating that a given pick belongs to the same event as the root pick (the first pick of the sequence, Y_0), and a value of 0 indicating that the two picks are unrelated. A final sigmoid activation function is applied separately to each hidden state output.
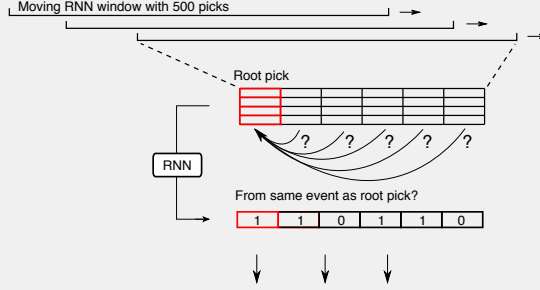
We apply the network to a sliding window of picks by incrementing over the entire sequence, shifting the window by one pick at a time. For the remainder of this paper, we refer to a fixed length sliding window of n_p picks as a *sub-sequence*. Here we use $n_p = 500$. After predictions have been made for a sub-sequence we drop the root pick and take the next pick as the root for the new sub-sequence. For each root pick we obtain a set of binary predictions about which of the following picks in the sub-sequence are related to the root.

Each of the picks in a sub-sequence is characterized by five input features, resulting in an input feature set with dimensions $(n_p, 5)$. The first two features are the latitude and longitude coordinates of the station that the pick was made on, which are both normalized to be in the range [0, 1] such that
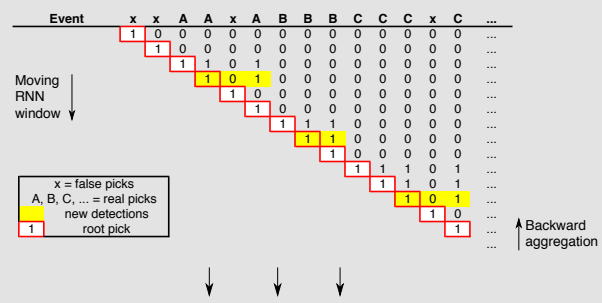
1. Collect picks over a seismic network

Time

Station latitude

Station lat.	33.57	33.56	33.57	33.49	...	33.49	33.59	33.35
Station lon.	-116.67	-116.53	-116.22	-116.60	...	-116.60	-116.76	-116.56
Time (sec)	4.56	4.84	4.93	5.11	...	5.23	5.62	5.65
Phase type	P	P	S	P	...	S	P	S

2. In moving window, predict which picks are from same event as root

Moving RNN window with 500 picks

Root pick

RNN

? ? ? ? ?

From same event as root pick?

1	1	0	1	1	0

3. Aggregate predictions for all windows

Event   x   x   A   A   x   A   B   B   B   C   C   C   x   C   ...

Moving RNN window ↓

x = false picks
A, B, C, ... = real picks
(yellow) new detections
1 (red) root pick

Backward aggregation

4. Pick sequence is fully associated
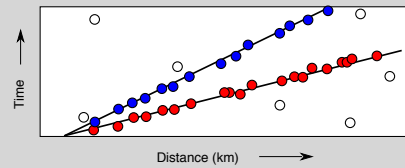
Time

Distance (km)

Figure 2: Overview of PhaseLink algorithm. A sliding window of picks is iteratively presented to a RNN, which outputs a binary sequence of equal length for each window. These output sequences indicate which picks (if any) are from the same event as the first pick in the window. Each pick in the sequence has five features: latitude, longitude, arrival time, phase type, and a binary padding indicator. The results from all windows are then aggregated to determine distinct clusters of picks (earthquakes detected).

the range spans the full dimensions of the seismic network. The third feature is the time of the pick, which is defined relative to the root pick within the sub-sequence. Here, we normalize the time values by a pre-defined maximum allowed value for picks to be included in a sub-sequence, which is chosen to be 120 seconds. This value is somewhat arbitrarily chosen, but ends up being not too important. The normalization ensures that this feature does not bias the training process. We discard any picks within the sub-sequence that are larger than 120 s, and pad the remainder of the feature window with zeros. The value of 500 picks is chosen loosely to correspond to the maximum number of picks that we expect to have within any 120 s window, which could vary depending on the problem. The penultimate feature is a binary value indicating the phase type, where a value of 0 means a P-wave,

Table 1: Model architecture

Layer	Neurons	Activation
Bidirectional GRU	200	sigmoid/tanh
Bidirectional GRU	200	sigmoid/tanh
Dense	1	sigmoid

and a value of 1 means an S-wave. Lastly, we have another binary indicator variable for whether a given pick is a zero-padded placeholder.

4.2 Aggregating Predictions

We now describe the final stage of PhaseLink, where the link predictions from each sub-sequence are aggregated to formally detect earthquakes. The output of the RNN is a prediction matrix that describes the link between each pick of a sub-sequence and its root pick (Figure 2, step 3). In order to assign picks to individual events, rather than to sub-sequence root picks, we cluster linked picks by incrementing backwards over the prediction matrix. This is performed as follows:

- For each sub-sequence, a cluster nucleates if at least n_{nuc} picks have predicted labels of 1.
- Once a cluster has nucleated, the set intersection is separately determined between it and every existing cluster.
- The existing cluster with the most picks in common is identified, and if this number is greater than n_{merge}, the two clusters are merged.
- After performing these steps for all sub-sequences, each remaining cluster is retained if the cluster size is at least n_{min} picks.

In this paper, we use $n_{nuc} = 8$, which was chosen to maximize the detection performance for the datasets used herein; varying this hyperparameter in the range 4-8 leads to relatively little change in performance on these datasets. Since the root pick is always linked to itself, the largest possible value of $n_{merge} = n_{nuc} - 1$. Here, we use this maximum value, $n_{merge} = 7$. Since merging clusters is performed by identifying the existing cluster with the most picks in common, there is the possibility that a pick could end up in two separate clusters; however in our testing, this is extremely uncommon. n_{min} is the most sensitive of the three hyperparameters, and its effect on the performance is examined in detail in the next section.

After applying the aforementioned steps, the PhaseLink algorithm is completed and the sequence is fully associated. We note that no hypocenters have been determined during this process for any events, whereas other associators jointly solve for a location as part of the detection process, which is a more challenging problem. Thus far, we have only discussed the method and how it is to be used; in the next section, we describe a scheme for generating the training data for the RNN.

4.3 Sim-to-Real Training

The RNN described in Section 4.1 could, in principle, be trained with real seismic phase data. However, even in the most seismically active regions, the available data may barely be enough to effectively train such a deep network. Since the network only requires phase arrival times and station geometries we can instead generate synthetic training datasets of arbitrary size. The key intuition here is that the supervised clustering problem solved by PhaseLink need not require fully realistic earthquake data to train an accurate predictor.

4.3.1 Synthetic Data Generation

We develop a simple scheme for generating large datasets of synthetic pick sub-sequences using a 1D layered model. The goal is for the neural network to learn the essential physics of wave propagation from the synthetic data, so that this knowledge can be directly applied to real data. To do so we define a set of rules from which random pick sub-sequences are generated. We use uniform distributions for all random quantities and denote this distribution as U. The rules to generate a single sub-sequence realization are as follows:

6

1. The initial number of events is chosen from $U \sim [0, 20]$.

2. A random hypocenter is initially assigned to all events. The latitude and longitude are each drawn from $U \sim [0, 1]$, while the depth is drawn from $U \sim [0, 25]$ km.

3. At a probability of 10%, each event is then separately reassigned a new hypocenter to produce events that overlap in time, but originate in different parts of the network.

4. The first event is assigned an origin time from $U \sim [-60, 60]$ s, enabling the possibility of the event to have an origin time before the sub-sequence starts.

5. The origin times for all subsequent events are chosen such that the time between consecutive events is $U \sim [3, 20]$ s

6. The maximum source-reciever distance for each event is drawn from $U \sim [20, 100]$ km.

7. Arrival times are calculated with a 1D model for all source-receiver combinations within the chosen maximum distance.

8. Picks are randomly discarded with $\mathrm{Pr} = 0.5$ to add variability to the station distribution.

9. Arrival time errors are added to each pick and drawn from $U \sim [-0.5, 0.5]$ s.

10. A random number of false picks drawn from $U \sim [0, 500]$ are randomly distributed across the network, with origin times drawn from $U \sim [0, 120]$ s.

11. Picks outside of the time window of $[0, 120]$ s are removed.

12. All remaining picks are sorted in time and the first 500 are retained. If fewer than 500 exist, the feature matrix is padded accordingly.

The ability to generate synthetic training data for the model to learn from has several important advantages. First, since a simple 1D layered velocity model is used to calculate arrival times, the method is easily applied to most tectonic regimes with relatively little knowledge required about the velocity structure. Second, errors can be directly added to the synthetic arrivals such that the model learns to deal with them in a rational way when examining real data. Third, an infinite amount of training data can be generated, which can prevent overfitting and regularization issues during the training process.

We produce two separate training datasets in this study. The first is for southern California, using the exact station distribution from the 811 past and present stations of the SCSN (Figure 3). The 1D velocity model used is from Hadley and Kanamori [9]. The second is for southwestern Japan, using 88 stations of the Hi-net seismic network (Figure 4). For the Japanese dataset, we use the 1D model of Shibutani et al [26].

Examples of two sub-sequence realizations are shown in Figure 5. The labels of each sub-sequence, Y_i, depend on whether the first pick, Y_0, is associated to an earthquake or not. If it is, then it and all picks associated with the event are given a label of 1, while all other picks are given a label of 0. Otherwise, Y_0 is the only non-zero value. We then repeat all of these steps 12 million times to generate a total of (up to) 6 billion picks.

4.3.2 Training the RNN

Given the generated datasets, we can train the RNN using any off-the-shelf machine learning package. The 12 million sub-sequences are designed to represent a wide variety of possible phase arrival time scenarios from all over southern California and Japan. From here, we can train the RNN to link phases together. We randomly split our 12 million sub-sequences into training (75%) and validation (25%) sets. We use a cross-entropy loss function and three NVIDIA GTX 1060 GPUs to train the model in mini-batches of 96 using the Adam optimization algorithm [13]. On the synthetic validation data, the model achieves classification performance of 99.92%. The training and validation loss histories are shown in Figure 6.

5   Results

In this section, we examine the performance of PhaseLink under a variety of scenarios. All of the tests are conducted in a controlled manner, such that ground truth is known for every single pick. This enables a detailed assessment of the performance at the individual phase level for real sequences of
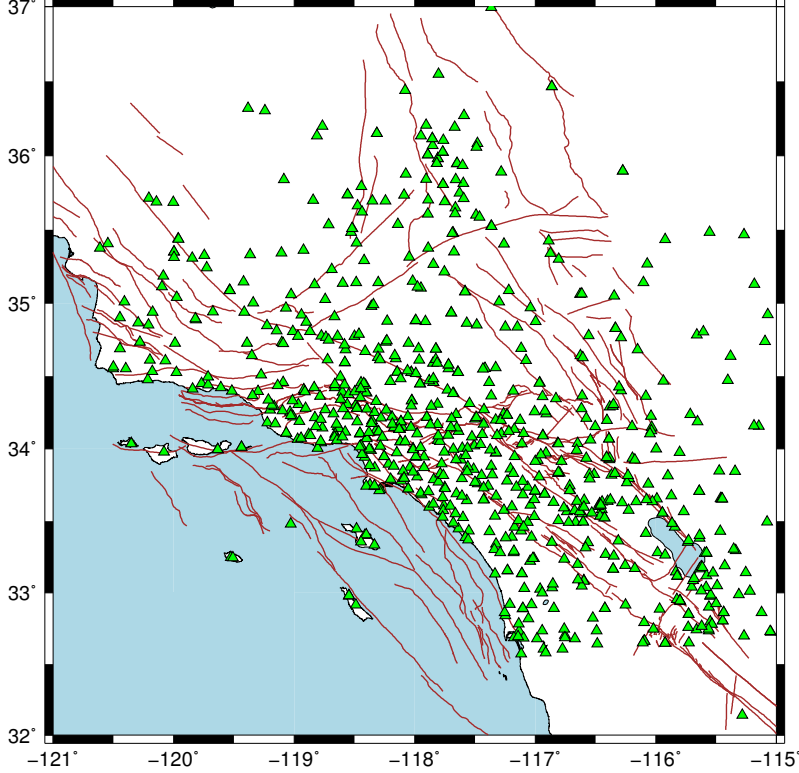
Figure 3: Map of southern California and the SCSN station distribution. The entire region shown defines the boundaries for generating training data.

picks, as well as sequences designed to test the point at which the method breaks down. It furthermore allows for a rigorous direct comparison of PhaseLink with existing grid association methods.

5.1 Application to 2016 Borrego Springs sequence

We apply PhaseLink to the 2016 Borrego Springs sequence [19], which occurred in the San Jacinto fault zone in southern California. For a controlled testing environment, we reconstruct the actual sequence of 73,353 picks (see Data section), which correspond to 1708 earthquakes that are listed in the SCSN catalog. Then we add in an equivalent number of false picks (73,353) uniformly distributed over the seismic network in time, i.e., picks that do not belong to an earthquake. This results in 50% of the picks in the sequence being false, but the effect is not uniform over time since the number of events (and therefore the number of picks) decreases with time after the mainshock. This has the overall effect of mimicking a real seismic network, where the system is dominated by real picks during a swarm, and later dominated by false picks the rest of the time.

First, we examine the performance of the neural network alone, without the clustering step included (Table 2). Precision is defined as the ratio of true positives to the true positives plus false positives. Recall is defined as the ratio of true positives to the true positives plus false negatives. The high precision for false picks (i.e. with true label = 0) shows that it is relatively rare for the network to assign a false pick label to real picks. Similarly, the high recall suggests that false picks are rarely assigned label 1. For the real picks (true label = 1), the high precision implies that the network rarely incorporates false picks into sequences of real picks. The lower recall, on the other hand, implies that it quite often discards real picks as false. Since the number of unrelated picks is much larger than that of related picks, these mis-classifications affect the real pick recall much more than the false pick precision.

However, this performance is only considering association relative to the root picks of individual sub-sequences. As we will demonstrate below the clustering scheme described in section 3 successfully
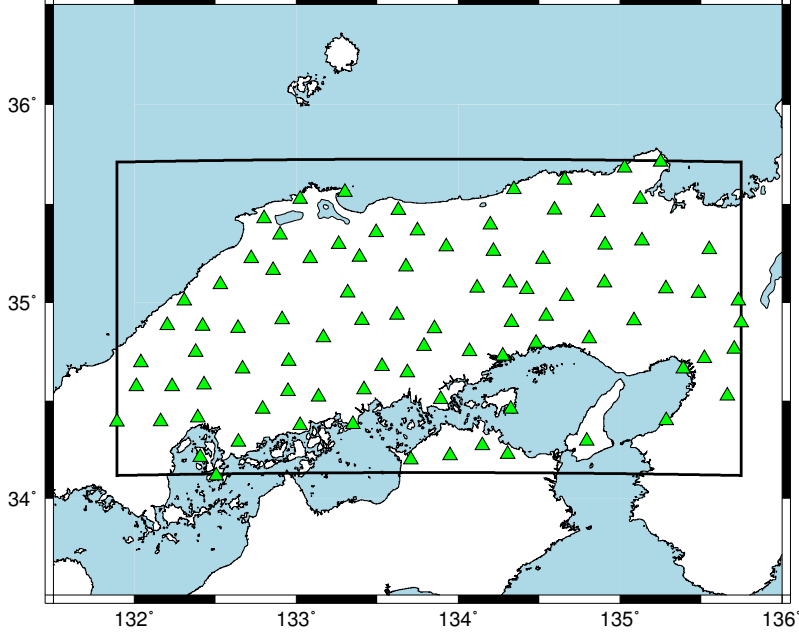
8

Figure 4: Map of southwestern Japan and station distribution (green triangles). The region used for generating synthetic events is indicated by the solid black line.

Table 2: RNN performance on validation dataset (individual phases)

True label	Precision	Recall	# samples
0	0.99	0.99	71116302
1	0.98	0.96	2236698

recovers many of these missed picks, since a pick only needs to be associated correctly in one of the sub-sequences it appears in.

Figure 7 demonstrates the outstanding performance of the complete algorithm in the context of detecting earthquakes, rather than individual phases. This precision-recall curve illustrates the inherent trade-off when the minimum number of picks per cluster, n_{min}, is varied. To determine whether the kth cluster of picks, A_k, corresponds to a successful event detection, we define the Jaccard precision between it and all clusters of picks in the ground truth, B_i,

$$J_k^p = \max_{i=1}^{c} \frac{A_k \cap B_i}{A_k \cup B_i}. \tag{1}$$

Here, c is the total number of events in the ground truth. If $J_k \geq 0.5$, we consider the detection successful. This means that at least 50% of the picks in the predicted cluster are common with a single event in the ground truth.

For all values of n_{min}, the precision is >0.996. However, raising n_{min} decreases the recall from 0.956 to ultimately 0.891, because real clusters that have fewer than n_{min} picks get discarded. High values of n_{min} decrease the algorithm's ability to detect and associate weakly recorded small earthquakes for which only small numbers of phase detections are available. However, at least for this test sequence, it appears that $n_{min} \sim 8$ is sufficient to get excellent association performance from the algorithm.

These performance numbers are in terms of event declarations, but it is also possible to evaluate the performance of phase associations as well. To do this, we calculate the mean value of J^p over the d detected events,

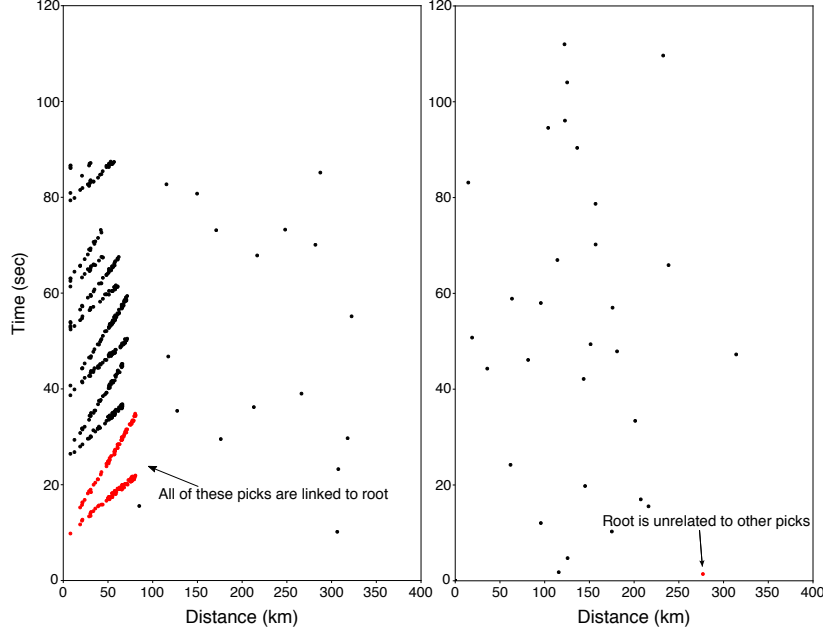$$\bar{J}^p = \frac{1}{d} \sum_{k=1}^{d} J_k^p. \tag{2}$$

9

Figure 5: Examples of synthetic training sub-sequences. Red picks are linked to the root, while black picks are unrelated to the root.

We further define the Jaccard recall as,

$$J_i^r = \max_{k=1}^{d} \frac{A_k \cap B_i}{A_k \cup B_i}, \tag{3}$$

and calculate the mean value of J^r for the dataset:

$$\bar{J}^r = \frac{1}{c} \sum_{i=1}^{c} J_i^r. \tag{4}$$

Together, $\bar{J}^p$ and $\bar{J}^r$ represent the precision and recall at an individual phase level, rather than an event level. These values are also shown in Figure 7 for the same range of n_{min} values, indicating that not only is the method detecting events well, but that it also reliably associates phases.

In developing a new method, it is also important to benchmark its performance against that of existing methods. Here, we compare the performance of PhaseLink against the grid associator, *dbgrassoc*, from the Antelope Environmental Monitoring Software package (BRTT Inc.). *dbgrassoc* is currently used by real-time seismic networks around the world, as well as researchers working with previously collected datasets in an offline mode. The program uses a pre-defined travel time grid that is set up over the region in which earthquakes are to be detected. There are a number of sensitive hyperparameters that control the detection process including the minimum number of picks, whether S-waves are to be included and how to deal with them, travel time residual limits, and the clustering time window. Once an event is detected, it also re-examines previous detections to see if the new event should be merged with another, or extra phases can be added in. For this comparison, we use the exact settings employed by a detection study in the San Jacinto fault zone [22], which are very similar to those used internally for real-time operation by the Anza seismic network. This ensures that *dbgrassoc* is correctly calibrated and that the comparison is fair.

We applied *dbgrassoc* to the same sequence of picks as used with PhaseLink, and the results are shown in Figure 7. At an event level, *dbgrassoc* and PhaseLink have nearly identical precision (>0.996), but the recall for PhaseLink is significantly higher (0.956 vs 0.919). When considering phase association performance, rather than event detection performance, *dbgrassoc* has slightly higher precision (0.976 vs 0.9718). However, PhaseLink has a much higher recall of 0.955, whereas *dbgrassoc* has a value of
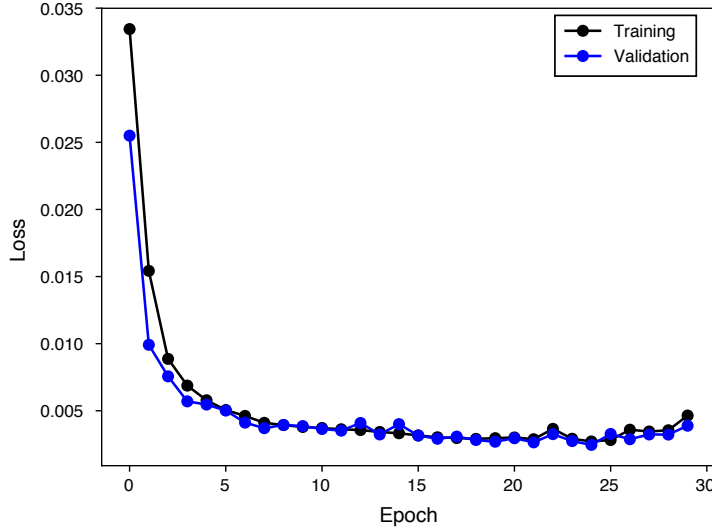
10

Figure 6: Training and validation loss histories.

0.904. Together, these tests show that PhaseLink detects significantly more events in the sequence and correctly associates more phases to each event, without sacrificing precision.

It should be noted that while this is a challenging sequence, the events were all large enough for humans to manually pick them. As seismologists seek to detect increasingly smaller magnitude events, not only will humans be unable to pick many of them, but the volume of data may be an order of magnitude larger [25, 21], with the timing between events significantly shorter. It is here that the full potential of PhaseLink is realized as compared with grid associators.

5.2 Stress testing PhaseLink in Japan

We next run PhaseLink through a series of tests to examine the conditions under which the algorithm finally breaks down. To do this, we construct random sequences of earthquakes synthetically with progressively more difficult scenarios: in each test, the average time between events is shortened relative to the previous test. We use a subset (88) of the Hi-Net seismic network stations from Japan that are concentrated around the Chugoku region (Figure 4). We generate 8 random sequences of 5000 earthquakes each within the region. For each sequence, the time between consecutive events is randomly drawn from a uniform distribution, with a fixed minimum value of 0 seconds, and a maximum value of 10, 12, 16, 20, 24, 32, 64, and 128 seconds, respectively. We then define $\overline{\Delta t_o}$ as the average time between events over the 5000 events in the sequence. Thus, in the hardest sequence, $\overline{\Delta t_o} = 5$ s, and in the easiest sequence, $\overline{\Delta t_o} = 64$ s. Each event is given a random hypocenter within the box defined by Figure 4, and the maximum source-receiver distance is drawn from $U \sim [20, 100]$ km. Arrival times are calculated with a 1D model for the region [26], and there are approximately 100,000 picks in each of the sequences. Since the minimum time between events is 0 for all sequences, this allows for the possibility that some events are simultaneous in time, but with different hypocenters, which is an extremely challenging problem. We then add random errors to the picks from $U \sim [-0.5, 0.5]$ s. Then, we run PhaseLink on the 8 sequences and calculate the performance against the ground truth for each one.

Figure 9 shows the results of the stress test on the synthesized Japanese data. For the easiest test, when $\overline{\Delta t_o} = 64$ s, the event precision and recall are 0.996 and 0.911 respectively. For the most difficult test, when events are 4 seconds apart on average, the performance is very poor, with the precision dropping to 0.603, and recall having a value of only 0.197. The figure indicates that the method stops achieving reasonable performance when the events are approximately 10-12 seconds apart, although this is somewhat subjective as there is not a steep falloff point in the curves. Regardless, a sequence of 5000 events that are 10 seconds apart on average is a very extreme scenario, and demonstrates
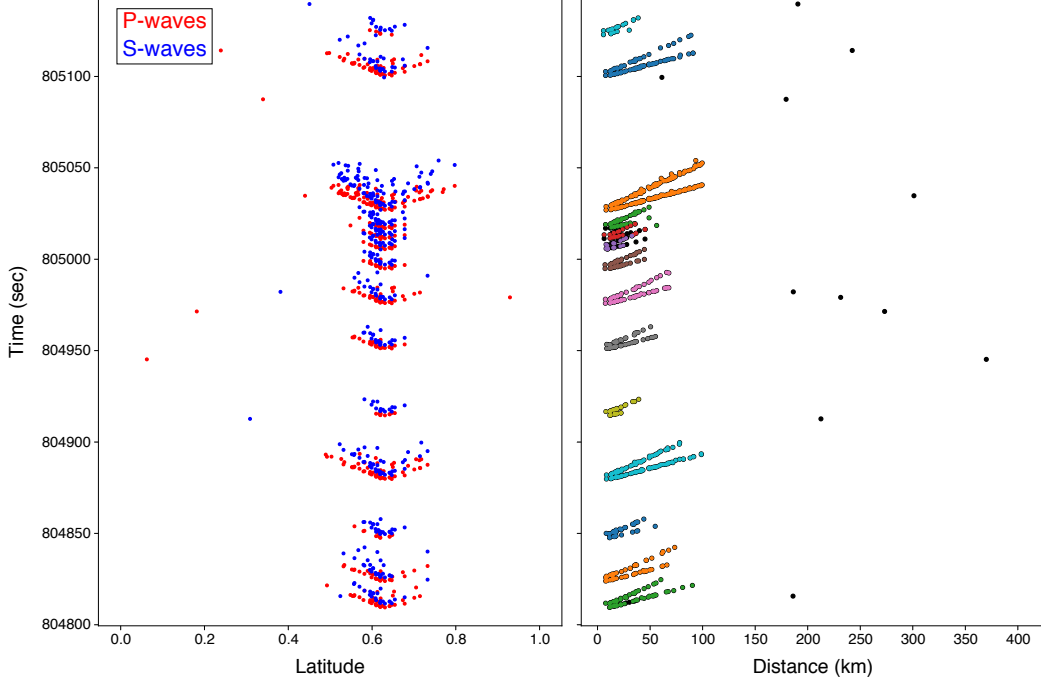
11

Figure 7: Sample of events detected during the 2016 Borrego Springs sequence. Within a span of 350 seconds, there are 14 events detected successfully. Left panel shows picks across the entire network for a given window of time. Latitude is normalized to be in the range (0, 1). Right panel shows output events detected and respective phases associated to each event (colored circles). Black circles indicate phases that are left unassociated by the algorithm.

the impressive performance of PhaseLink as an association algorithm. We also performed the same tests with *dbgrassoc*, and the performance is shown in Figure 9 as well. It is clear that even for the easiest test that *dbgrassoc* missed nearly 40% of the events (compared with 9.5% for PhaseLink) and had a false positive detection rate that is nearly 10 times higher than PhaseLink. This is true at a phase association performance level as well. We conclude that PhaseLink significantly outperforms *dbgrassoc* when events are densely clustered in time.

6   Discussion

Earthquake phase association is an essential task in seismology that has been around for decades but still remains a challenge at the present day, most noticeably during dense earthquake sequences. PhaseLink is distinctly different from previous methods for phase association because it does not need to search over a hypocenter grid for a solution. Instead, it uses the capabilities of deep neural networks to learn the patterns that seismic waves make as they propagate across a seismic network. The results indicate that the method has all of the key functionality necessary to solve this difficult problem in an efficient and robust way, and that recurrent neural networks are capable of learning the basic physics of seismic wave propagation in the Earth. Here, we discuss several important aspects about the algorithm in more detail.

6.1   Potential future modifications

It is reasonable to expect that some modifications to the methodology will be necessary in the future. What distinguishes PhaseLink from other association algorithms is that when persistent problems are identified, examples of such cases can be added to the training dataset so that the neural network learns how to deal with them. This is a fundamentally different approach from grid-based association
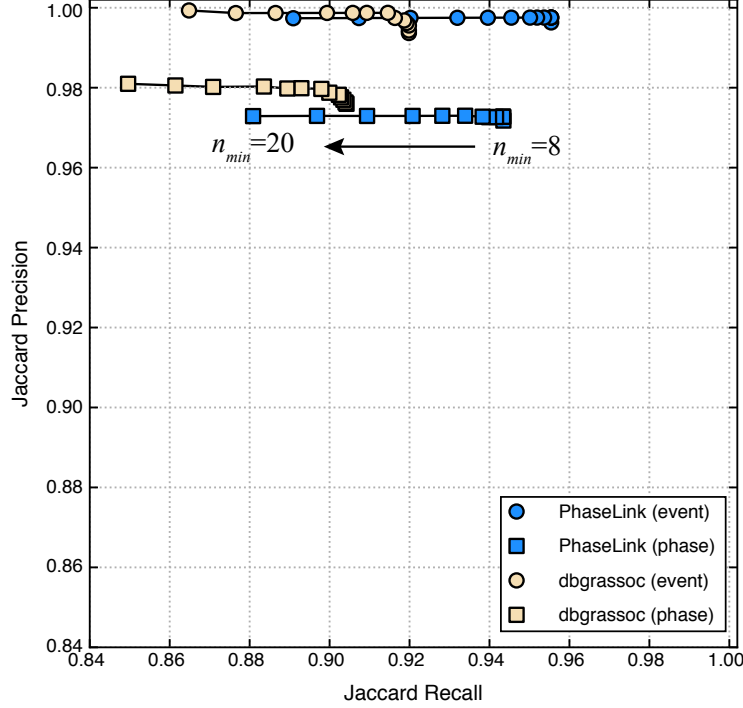
12

Figure 8: Precision-recall tradeoff curve for event detection and phase association performance on the 2016 Borrego Springs sequence. Each point represents a value of n_{min} from 8-20.

algorithms, where addressing problematic cases must be done by either tuning hyperparameters or modifying the algorithm altogether.

We have demonstrated the usage of PhaseLink only in the context of a local seismic network. Applying the method to regional or teleseismic scales may require additional modifications, but we do not see anything that would preclude the method from being used at these scales. At such distances, there are often more phase types available that could be explicitly incorporated into the problem. Since PhaseLink automatically associates phase picks together without solving for a location, once a hypocenter is determined, it is possible to then perform a quick back-projection of all nearby phases to clean up or add in any split events or unassociated phases.

6.2   Comparison of PhaseLink and Grid Associators

Grid associators link phases together while simultaneously locating an event. As this is essentially solving two (interdependent) problems, it can lead to false event detections if a random subset of phases have a moveout that is consistent with a single grid node acting as a hypocenter. Furthermore, the large number of ad hoc hyperparameters that are necessary to stabilize the association scheme can lead to various types of errors. PhaseLink does not use a grid or locate earthquakes; rather, it solves a simpler problem: group phases together with arrival time patterns that are generally consistent with seismic wave propagation, irrespective of the hypocenter. This will enable robust phase association, and these phases can then be presented to a location algorithm to determine the best possible hypocenter.

6.3   Advantages of training with synthetic data

A key advantage of the algorithm is that it can be trained using only synthetic data. Because of this it is directly applicable to regions for which there is insufficient real phase data. This includes small networks as well as temporary ones that have never recorded data before. The only requirement is a sufficiently accurate velocity model for generating the training data, which for most cases, is probably a 1D layered model.
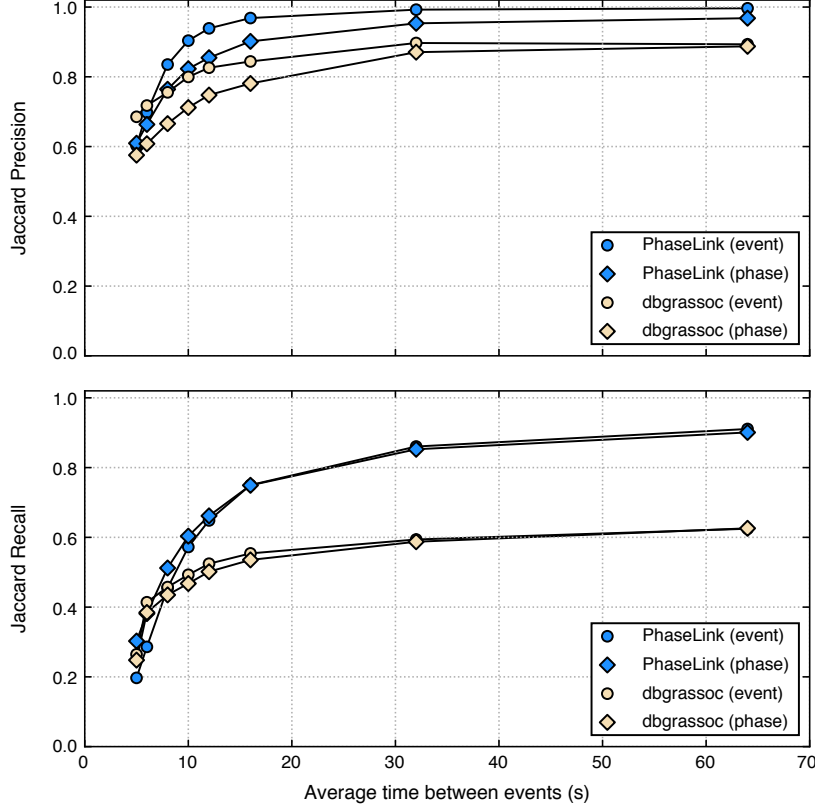
Figure 9: Stress testing PhaseLink and dbgrassoc on synthetic earthquake sequences in Japan. As the average time between events in a sequence decreases, the precision and recall decrease for both methods. PhaseLink significantly outperforms dbgrassoc even on the easiest cases in the test dataset. These tests indicate that PhaseLink begins to break down when the average time between events approaches 10-12 seconds.

Another distinct advantage of PhaseLink is that picking errors can be explicitly accounted for in the training data. Here, we chose a rather extreme case by drawing errors from $U \sim [-0.5, 0.5]$ s to force the neural network to learn how to deal with this complexity. This also has the advantage of learning to rationally deal with genuine perturbations in arrival times due to 3D structure. We note that in a limited sense, this attribute is similar to the BayesLoc algorithm [17], which uses a Bayesian formalism to account for the possibility of errors in phase picks and other factors for association purposes.

The developed algorithm has the potential to significantly improve automated processing of large seismic datasets. This is important both for researchers who are looking to build catalogs from long-existing or newly collected datasets, as well as seismic networks that are responsible for routine seismic monitoring. It further has applications to microseismic monitoring, and to earthquake early warning. The latter is a particularly exciting application, because if the long-term memory abilities of RNN are fully utilized, wave propagation patterns made by foreshocks could be learned from dynamically and used by the RNN to make a high-confidence event detection of a large magnitude event with only one or two stations.

7 Conclusions

We have developed a new method for performing seismic phase association using deep learning. The method has been shown to achieve outstanding performance on an extremely active aftershock sequence in southern California, where many of the events are only seconds apart in origin time.

It can be trained using only synthetic seismic phase arrival time data, and can therefore be applied even to networks that lack large amounts of labeled training data. The method does not use grids in any form to solve the association problem – it instead learns to scan sequences of picks and identify patterns that resemble those of seismic wave propagation in the Earth.

Acknowledgements

References

[1] R. Allen. Automatic phase pickers: Their present use and future prospects. *Bulletin of the Seismological Society of America*, 72(6B):S225–S242, Dec. 1982.

[2] R. V. Allen. Automatic earthquake recognition and timing from single traces. *Bulletin of the Seismological Society of America*, 68(5):1521–1532, Oct. 1978.

[3] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, J. Chen, J. Chen, Z. Chen, M. Chrzanowski, A. Coates, G. Diamos, K. Ding, N. Du, E. Elsen, J. Engel, W. Fang, L. Fan, C. Fougner, L. Gao, C. Gong, A. Hannun, T. Han, L. Johannes, B. Jiang, C. Ju, B. Jun, P. LeGresley, L. Lin, J. Liu, Y. Liu, W. Li, X. Li, D. Ma, S. Narang, A. Ng, S. Ozair, Y. Peng, R. Prenger, S. Qian, Z. Quan, J. Raiman, V. Rao, S. Satheesh, D. Seetapun, S. Sengupta, K. Srinet, A. Sriram, H. Tang, L. Tang, C. Wang, J. Wang, K. Wang, Y. Wang, Z. Wang, Z. Wang, S. Wu, L. Wei, B. Xiao, W. Xie, Y. Xie, D. Yogatama, B. Yuan, J. Zhan, and Z. Zhu. Deep speech 2 : End-to-end speech recognition in english and mandarin. In M. F. Balcan and K. Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 173–182, New York, New York, USA, 20–22 Jun 2016. PMLR.

[4] M. Bilenko, S. Basu, and R. J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *Proceedings of the twenty-first international conference on Machine learning*, page 11. ACM, 2004.

[5] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv:1406.1078 [cs, stat]*, June 2014. arXiv: 1406.1078.

[6] P. M. R. DeVries, F. Viégas, M. Wattenberg, and B. J. Meade. Deep learning of aftershock patterns following large earthquakes. *Nature*, 560(7720):632–634, Aug. 2018.

[7] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.

[8] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[9] D. Hadley and H. Kanamori. Seismic structure of the Transverse Ranges, California. *GSA Bulletin*, 88(10):1469–1478, Oct. 1977.

[10] S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, Nov. 1997.

[11] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, Apr. 1982.

[12] C. E. Johnson, A. Bittenbinder, B. Bogaert, L. Dietz, and W. Kohler. Earthworm: A flexible approach to seismic network processing. *Iris newsletter*, 14(2):1–4, 1995.

[13] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, Dec. 2014. arXiv: 1412.6980.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[15] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015.

[16] P. Lu, M. Morris, S. Brazell, C. Comiskey, and Y. Xiao. Using generative adversarial networks to improve deep-learning fault interpretation networks. *The Leading Edge*, 37(8):578–583, Aug. 2018.

[17] S. C. Myers, G. Johannesson, and W. Hanley. A Bayesian hierarchical method for multiple-event seismic location. *Geophysical Journal International*, 171(3):1049–1063, Dec. 2007.

[18] T. Perol, M. Gharbi, and M. Denolle. Convolutional neural network for earthquake detection and location. *Science Advances*, 4(2):e1700578, Feb. 2018.

[19] Z. E. Ross, E. Hauksson, and Y. Ben-Zion. Abundant off-fault seismicity and orthogonal structures in the San Jacinto fault zone. *Science Advances*, 3(3):8, Mar. 2017.

[20] Z. E. Ross, M.-A. Meier, and E. Hauksson. P Wave Arrival Picking and First-Motion Polarity Determination With Deep Learning. *Journal of Geophysical Research: Solid Earth*, 123(6):5120–5129, June 2018.

[21] Z. E. Ross, M.-A. Meier, E. Hauksson, and T. H. Heaton. Generalized Seismic Phase Detection with Deep Learning. *Bulletin of the Seismological Society of America*, 2018.

[22] Z. E. Ross, M. C. White, F. L. Vernon, and Y. Ben-Zion. An Improved Algorithm for Real-Time S-Wave Picking with Application to the (Augmented) ANZA Network in Southern California. *Bulletin of the Seismological Society of America*, July 2016.

[23] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, Nov. 1997.

[24] A. Shafaei, J. J. Little, and M. Schmidt. Play and learn: Using video games to train computer vision models. In *BMVC*, 2016.

[25] D. R. Shelly, W. L. Ellsworth, and D. P. Hill. Fluid-faulting evolution in high definition: Connecting fault structure and frequency-magnitude variations during the 2014 Long Valley Caldera, California, earthquake swarm. *Journal of Geophysical Research-Solid Earth*, 121(3):1776–1795, Mar. 2016.

[26] T. Shibutani and H. Katao. High resolution 3-D velocity structure in the source region of the 2000 Western Tottori Earthquake in southwestern Honshu, Japan using very dense aftershock observations. *Earth, Planets and Space*, 57(9):825–838, 2005.

[27] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1297–1304. Ieee, 2011.

[28] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[29] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pages 23–30. IEEE, 2017.

[30] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. In *SSW*, page 125, 2016.

[31] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl, et al. Constrained k-means clustering with background knowledge. In *ICML*, volume 1, pages 577–584, 2001.

[32] D. Wang and J. Eisner. Fine-grained prediction of syntactic typology: Discovering latent structure with *supervised* learning. *Transactions of the Association for Computational Linguistics (TACL)*, 5:147–161, June 2017.

[33] Y. Wu, Y. Lin, Z. Zhou, D. C. Bolton, J. Liu, and P. Johnson. DeepDetect: A Cascaded Region-Based Densely Connected Network for Seismic Event Detection. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–14, 2018.

[34] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo. Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4651–4659, 2016.

[35] W. Zhu and G. C. Beroza. PhaseNet: A Deep-Neural-Network-Based Seismic Arrival Time Picking Method. *arXiv:1803.03211 [physics, stat]*, Mar. 2018. arXiv: 1803.03211.