

ESTIMATION OF SITE AMPLIFICATION FROM GEOTECHNICAL ARRAY DATA USING NEURAL NETWORKS

Daniel Roten and Kim B. Olsen

Department of Geological Sciences, San Diego State University

Abstract

We use deep learning to predict surface-to-borehole Fourier amplification functions (AFs) from discretized shear-wave velocity profiles. Specifically, we train a fully connected and a convolutional neural network (NN) using observed mean AFs observed at ~ 600 KiK-net and California Strong Motion Instrument Program (CSMIP) vertical array sites. Compared to predictions based on theoretical SH 1D amplifications, the NN results in up to 50% reduction of the mean squared log error between predictions and observations at sites not used for training. In the future, NNs may lead to a purely data-driven prediction of site response that is independent of proxies or simplifying assumptions.

Introduction

The densification of seismic networks, such as the CSMIP strong motion network in California and the KiK-net observatory in Japan, have vastly increased the number of earthquake records available for strong motion research. In addition, the deployment of borehole accelerometers at many locations has resulted in a large volume of vertical array data which has contributed to a better understanding of linear and nonlinear site response during strong shaking [e.g. 1–4].

However, despite the increased amount of data, the standard deviations of intensity measures in ground motion prediction equations (GMPEs) have barely decreased over the past four decades [5, 6]. Standard deviations in GMPEs remain high because empirical methods use very simple models to approximate highly complex wave propagation phenomena [7]. Site conditions in most GMPEs are typically reduced to the average velocity in the top 30 meters, V_{S30} and in some cases basement depth (e.g. the depth to a constant shear-wave velocity of 1 km/s, Z_1) [8]. Similarly, ground motions recorded on vertical arrays have demonstrated the shortcomings of current site response prediction techniques, in particular the assumption of a laterally constant medium [2, 9, 10]. Three dimensional simulations with sophisticated structural models and nonlinear wave propagation codes are needed to study the response of such sites [e.g., 11, 12]. Although such case studies may shed light on the wave propagation effects behind the site response observed at a particular location, it is not clear how this approach can be generalized to sites for which no sophisticated 3D velocity models are available. Clearly, new methods are needed which harness the sheer volume of strong motion data (including data acquired on vertical arrays) to improve seismic hazard analysis.

While seismology has always been a data intensive field, enormous amounts of data are

nowadays being collected in a broad spectrum of fields ranging from technology to finance to healthcare. Combined with increasingly powerful computers, the availability of these very large datasets has been driving progress in machine learning (ML) techniques, in particular deep learning applications which thrive under large amounts of data.

An exciting aspect of deep neural networks (NNs) is their ability to detect patterns in the input data which allows them to make sense of labeled output data. In contrast to shallow learning algorithms, deep neural networks are less dependent on feature engineering, i.e., the process of transforming input data into features from which the output can be derived using a simple mathematical expression. In site response prediction, one could think of proxies such as V_{S30} or Z_1 as engineered features needed to carry out regression analysis for calibration of traditional GMPEs (i.e., a shallow learning method). A deep learning algorithm would not depend on such engineered features, and could process the entire velocity information available for a site, without resorting to simplifications which discard valuable data. The idea is that the network will identify new features from the provided velocity profile, which guide a more accurate site response prediction compared to proxies such as V_{S30} or Z_1 .

In this study, we propose to train a deep neural network to learn the observed site response at CSMIP and KiK-net sites based on the entire soil stratigraphy and potentially other parameters characterizing the incoming wavefield. The goal is to develop a method which predicts site response based entirely on observed vertical array data, without relying on simplifying assumptions, such as one-dimensionality or a vertically-incident wave field, made in traditional site response assessment.

We have arranged the content of our paper as follows. In section 2, we provide a quick overview of deep learning and elaborate on the design of the NN. We have tested the feasibility of the method by calculating theoretical, one-dimensional SH transfer functions (SH1D) for 662 real KiK-net soil profiles, and for training a fully connected neural network to predict the amplification functions (AFs) from the soil profiles. In second section 3, we describe the data preparation methods used in the calculation of transfer functions from CSMIP geotechnical arrays and KiK-net strong motion sites. Results of site response prediction using the deep learning are presented in section 4.

Deep Learning and Design of Neural Networks

Artificial neural networks (ANNs) are modeled after biological neural networks found in animal brains, and consist of a collection of artificial neurons interacting with each other. ANNs are typically organized in layers, and every ANN consists of an input layer accepting the input parameters and an output layer which produces the desired prediction.

Overview of Neural Networks

Deep neural networks feature at least one, but typically several hidden layers located between the input and output layers. In a fully connected ANN, also called multi-layer perceptron (MLP), each artificial neuron in each layer is connected to every other neuron of the previous and next layer. (Fig. 1). Therefore, each neuron receives an input signal from every

neuron in the previous layer, and redirects a modified signal to every neuron in the next layer. The input function $z(\mathbf{x})$ into a neuron consists of a weighted sum of the inputs $\mathbf{x} = (x_1, x_2, x_3, \dots, x_m)$ from the m individual neurons of the previous layer:

$$z(\mathbf{x}) = \mathbf{w}\mathbf{x} + b,$$

where \mathbf{w} is a vector describing the weights of each neural connection. The offset b is also called bias. If the current layer has n nodes and the previous layer has m nodes, there are $n \times m$ connections from the current to the previous layer, and the weight matrix connecting the two nodes has shape $n \times m$. In addition, there are n offsets that must be trained. The weights \mathbf{w} and offset \mathbf{b} are trainable parameters.

Activation functions allow the ANN to learn nonlinear functions. Without activation functions, the total output of the ANN would represent a linear function regardless of the depth of the networks (i.e., regardless of the number of hidden layers). Typical choices of activation functions include sigmoids, rectified linear units (ReLU) or hyperbolic tangent functions (tanh) [13]. If the ANN is used for regression, the output node uses a linear activation, allowing the ANN to output any real number.

The weights and biases are optimized by training the ANN. The goal of training is to minimize the loss function, which quantifies the difference between the desired output provided in the training data and the network's actual output. Forward propagation in a feedforward ANN refers to the computation of the network's output value based on the chosen input and the ANN's current weights and biases, with information flowing from the input to the output layer. This order is reversed during backpropagation, where the gradient of the loss function with respect to the ANN's weight is computed based on the input and desired output of one or several training examples. Training consists in minimizing the loss by performing gradient descent on the loss function.

Because there are many trainable parameters in an ANN, and the number of training examples is often limited, deep neural networks are prone to overfitting [e.g. 13]. An overfitted model will perform very well on the input set but will generalize poorly to the test set, with low misfit error on the training set but high error on the test set (i.e., the model exhibits high variance). Overfitting also affects inversion problems encountered within different domains of seismology, such as seismic tomography [e.g. 14]. A common technique to reduce variance (overfitting) in such scenarios is to add L1 or L2 regularization, which penalizes large weights and thereby reduces the number of free parameters in the model. Although this type of regularization can also be applied to deep neural networks, it is more common to reduce variance using a technique called dropout [15]. In dropout regularization, a predefined fraction of neurons is randomly eliminated during each training iteration. This prevents the network from relying on a single feature, and allows it to generalize better to data it has not encountered during training.

FCNN Architecture

In the fully connected ANN design used in this study, the input layer expected the shear-wave velocities extracted at 100 predefined depths from the soil profile (Fig. 1a,c); the sampling

interval gradually increased from 1 m near the surface to 30 m at 1,500 m depth. The properties of the last layer were projected onto the remaining depth intervals at shallower sites. The desired frequency of site amplification was also provided to the ANN algorithm and represented the last value in the input layer. The output layer consisted of a single neuron with the site amplification value at the specified frequency (Fig. 1b,c). Our design chosen for the fully connected neural network (FCNN, Fig. 1c) used a many-to-one layout, accepting many inputs but producing just a single output value. That is, the ANN only predicted the amplification at one frequency at a time. One could also design a similar network using a many-to-many configuration, and predict the amplification at several frequencies at the same time. We experimented with both many-to-one and many-to-many designs and found that the many-to-one configuration was superior for the FCNN. However, a many-to-many design was adopted for the convolutional neural network (CNN) described below.

Our FCNN used 7 hidden layers, and the number of neurons decreased gradually from 256 nodes in the first hidden layer towards the single-node output layer. Following DeVries et al. [16], the activation function assigned to the hidden layers alternated between hyperbolic tanh and ReLUs; and a linear activation function was applied at the output layer (Fig. 1).

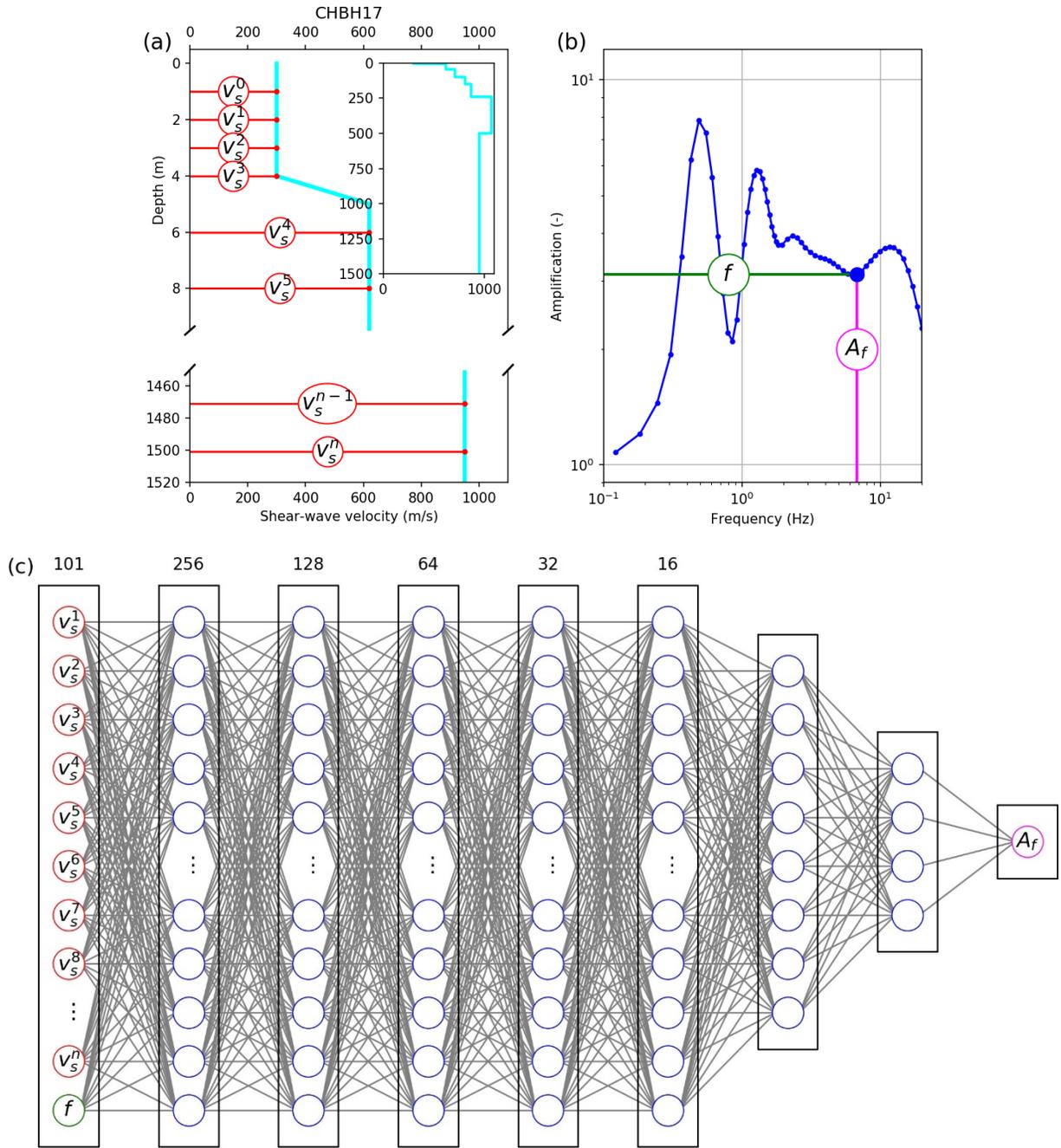


Figure 1: Prototype of ANN for prediction of simulated transfer functions. (a) Shear-wave velocities (v_s , red nodes) were discretized at $n = 100$ discrete depths and (c) fed into the input layer along with the frequency f of amplification (green node). Hidden layers in (c) are shown by blue neurons. Where not all nodes are shown, the true number of nodes are given at the top of the layer. The output node contains the amplification A_f at the specified frequency (b).

CNN Architecture

In a convolutional layer, nodes are not directly connected to nodes in the next layer. Instead, the data in the layer are convolved using a series of filters. The dimensions of the convolutional layer's output depend on the type of convolution (overlap and stepping size) and the number of filters, with each filter creating a new representation of the input data. However, as most convolutional layers use many filters, the output is typically large, and downsampled in a pooling layer following the convolutional layer. A set of convolutional and pooling layers may either be followed by another set of convolutional and pooling layers, or the data is flattened and directed into a fully connected layer.

CNNs are especially effective for image recognition or classification problems, as they are able to extract information from the spatial arrangement of the pixels. Although predefined filters have long been used in image processing, the effectiveness of CNN derives from the network's ability to optimize the filters depending on the training data. In other words, the filter parameters are optimized during back-propagation such that the features extracted by the different filters are effective at carrying out the CNN's task.

In our case, we applied a CNN to take advantage of the spatial information in the velocity profile. We used v_s and the P-wave velocity v_p as different image 'channels', analogous to the red, green and blue channels used in image recognition. In contrast to image recognition, where the input image is three-dimensional (two spatial dimensions plus three channels), our input was only two-dimensional (v_s and v_p at different depths). We did not use densities as they were not provided for KiK-net profiles.

In our CNN design, we used a single convolutional layer with 16 filters of dimensions 5 2 (Fig. 2) right after the input layer (dimension 1002, with v_s and v_p at 100 predefined depths). The output of the convolutional layer consisted of 100 16 values, which we reduced to 25 16 values using a pooling layer. The output of the pooling layer was flattened and fed into a fully connected layer of 512 nodes. Two more hidden layers with 256 and 128 nodes followed. The output layer contained 50 nodes, which represented the desired amplification function at 50 predefined frequencies (Fig. 2). A ReLU function was used for activation right after the pooling layer, and we alternated between ReLU and tanh functions in the three fully connected layers. As in the FCNN, a linear activation was used in the output layer. Dropout regularization was applied after each layer. Batch normalization (Fig. 2) was carried out before each activation to improve

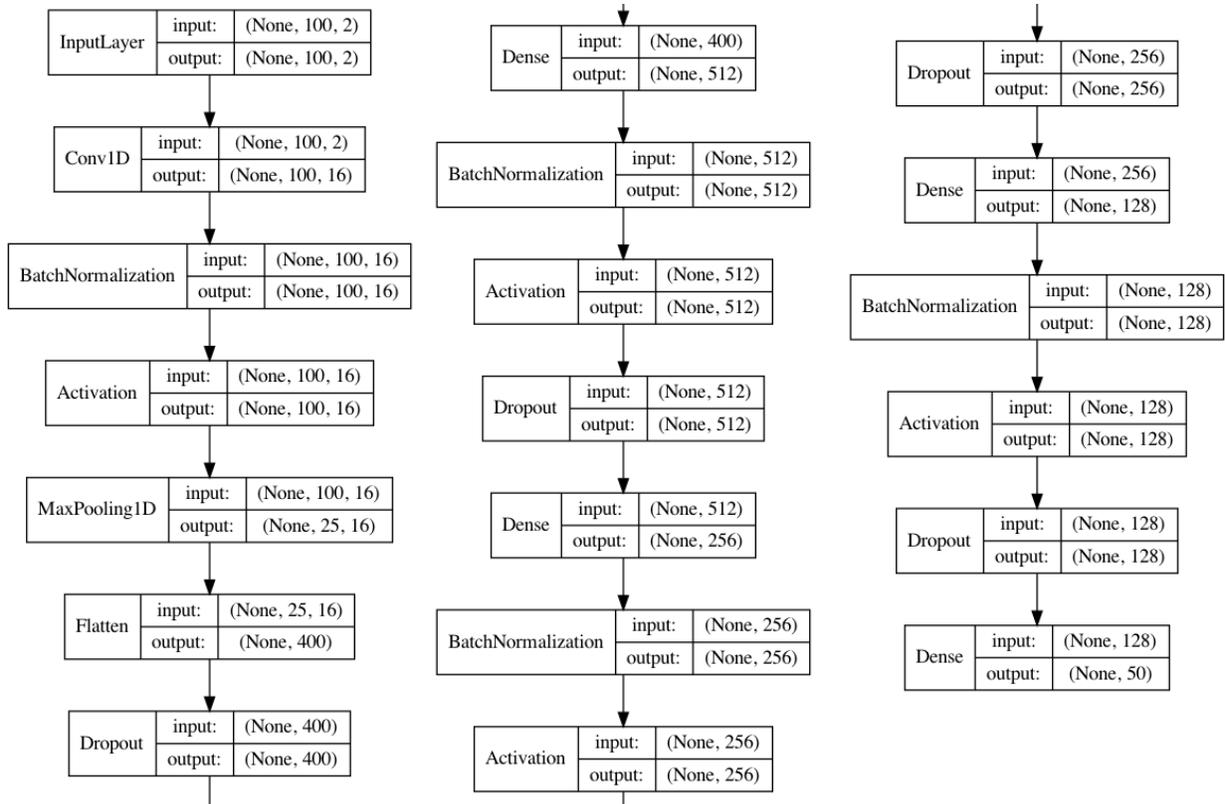


Figure 2: Architecture of convolutional neural network (CNN) for prediction of site response.

convergence (this was also done in the FCNN). Both the FCNN and the CNN were implemented with the Keras library for Python [17] using the TensorFlow[18] backend.

Data Preparation

Because data from CSMIP and KiK-net are stored in different formats, separate workflows were developed to extract transfer functions from CSMIP and KiK-net data, respectively.

CSMIP Data

Records from vertical arrays in the CSMIP network were downloaded from the Center for Engineering Strong Motion Data website [19]. In a first step, all earthquakes which resulted in peak ground accelerations (PGA) > 0.01g on geotechnical arrays within the CSMIP network were identified and retrieved from the network’s website. We were able to download these records directly from the web interface as the amount of data was relatively small. The search resulted in a total of 209 suitable records pertaining to 99 different local and regional earthquakes with magnitudes between 3.1 and 7.3 (Fig. 3). The retrieved records include 4 records of the M7.1 Ridgecrest earthquake of July 5, 2019 that were acquired on geotechnical arrays in Palmdale, Oxnard and Los Angeles. 41 out of the total 44 CSMIP geotechnical arrays are represented in these records.

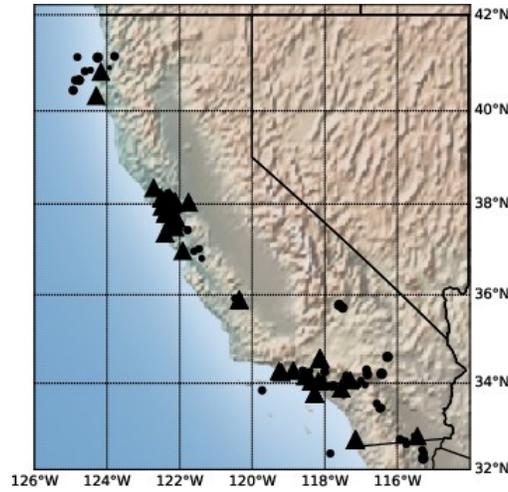


Figure 3: Location of CSMIP geotechnical arrays (triangles) and selected earthquakes (circles).

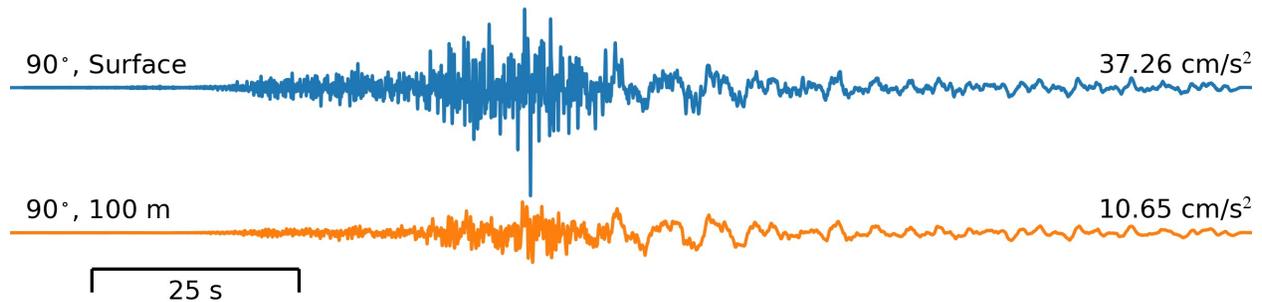


Figure 4: Surface and borehole accelerograms recorded at the La Cienega (Los Angeles) geotechnical array during the 2019 M 7.1 Ridgecrest earthquake.

A workflow was developed which parses the CSMIP ASCII files [20] to extract ground motions, performed Fast Fourier Transform, smoothing with a Konno-Ohmachi filter [21] and the computation of the AF. Figure 4 shows an example of surface and downhole accelerograms used to train the NN, namely records from the La Cienega (Los Angeles) geotechnical array during the M 7.1 Ridgecrest earthquake (rupture distance 205 km). Although the incoming wavefield is dominated by long-period ground motions (period T 10s) ground motions (Fig. 5a), the amplification by the deep alluvium can clearly be observed at frequencies between 1 and 10 Hz (Fig. 5b). The observed Fourier amplification at different frequencies obtained from the chosen records represents the output that the NN is trained to predict. Shear-wave velocities at different depths, along with selected earthquake parameters (e.g., magnitude, rupture distance) represent the input layer of the NN.

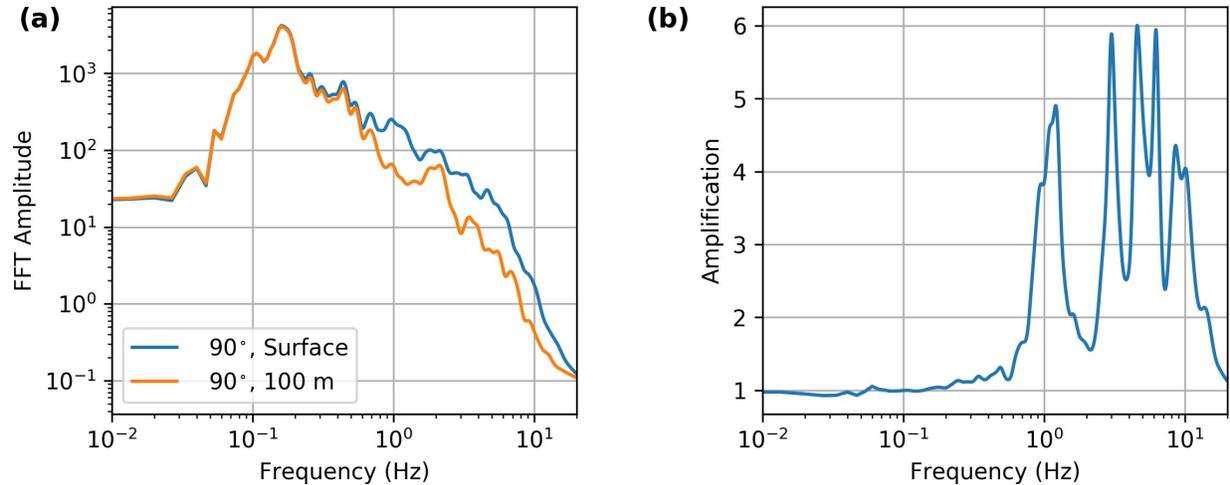


Figure 5: (a) Fourier amplitude and (b) AF (amplification function) from La Cienega records of *M* 7.1 Ridgecrest earthquake.

KiK-net Data

Although the KiK-net data website provides the option to search for earthquake records based on different parameters, we found that the interface was not suitable to select and download the relatively large amount of records that we wanted to use for this project. Instead, we developed a script that downloads all the earthquake records from the KiK-net website and stores them locally. This 'data scraper' was programmed in Python using the requestslibrary [22].

Acceleration time series from all earthquakes recorded by KiK-net stations between January 1997 and August 2020 were downloaded from the Kyoshin web site [23]. Five parallel download sessions were executed to retrieve the records, which amount to about 105 Gb in total. Records of acceleration time series in K-NET ASCII format were kept in event tar archives for all KiK-net sites which recorded the event. Earthquake and station metadata were extracted from event files and stored in a local database. Three separate tables with station information (i.e., station code, location, elevation, sensor depth), event information (event ID, date, magnitude, hypocenter) and record informations (event ID, station code, peak ground acceleration, distance) were generated and stored in Python Pandasdataframes.

Based on the record's PGA (peak ground acceleration), we selected 20 representative events for each station. Two different selection strategies were used to generate training datasets for prediction of mean amplifications or event-specific amplifications, described below.

For the prediction of mean site amplifications, we only used records with surface PGAs below 0.2g to exclude nonlinear effects. Where available, we randomly picked 20 events with PGAs within 0.05 and 0.2g. If less than 20 events with $0.05g < PGA < 0.2g$ were available, we selected the 20 events with the highest PGA. The number of 20 events per site was chosen because all except 4 sites (KNMH18, FKOH02, SOYH3, AICH23) recorded more than 20 events until August 2020, and no site recorded less than 10 events. Using more events per site would be

possible, but would require the introduction of training weights to give all sites equal consideration; otherwise the neural network would tend to automatically give more weight to stations that are represented by more entries in the training dataset.

For the prediction of event-specific site amplification, events were selected such that their PGAs were roughly uniformly distributed between a minimum PGA of 0.05g and the maximum PGA recorded at that site. If less than 20 events with $\text{PGA} > 0.05\text{g}$ were available at a site, we picked the 20 events with the highest PGAs. This approach was chosen to ensure sufficient representation of events with high PGAs for the prediction of nonlinear effects. However, results presented in this paper are focused on the prediction of mean AFs, which do not include nonlinearity.

The next step in the KiK-net data processing workflow consisted in the extraction of time series data from the selected observations. We used the ObsPyFramework for Python [24] to read the data. Surface-to-borehole transfer functions were computed for both horizontal components, smoothed using a Konno-Ohmachi filter (bandwidth $w = 10$) and the geometric mean of both horizontals was computed. We then interpolated the amplification at 50 frequencies of interest, which are logarithmically spaced between 0.3 and 20 Hz. This procedure was carried out for a total of 13,210 events. Computationally, the data preparation was expensive because two Fast Fourier Transforms and Konno-Ohmachi filtering operations were executed for each record. In order to accelerate the process, the web scraper and data processing workflow were deployed on the commodity cluster Rhea at the Oakridge Leadership Computation Facility (OLCF). We used the Apache Spark Engine [25] to distribute the data processing on up to 5 nodes and 80 CPU cores. This approach resulted in a wall-clock time of less than two hours for the computation and smoothing of the amplification functions for all the 13,210 records.

Training and test datasets were created as follows: First we randomly selected 95% of the sites to contribute to the training set, while the remaining sites were assigned to the test set. Figure 6 shows the distribution of training and test sites among the KiK-net stations. We used the same selection of training and test sites for all different neural network layouts and hyperparameter choices shown in this paper, in order to allow for a one-by-one comparison of network performances. We created training and test datasets by iterating over all the records pertaining to each given training and test site. In the many-to-many layout used in the convolutional neural network for the prediction of mean AFs, the training and test sets contained just one data point per site. In the many-to-one design of the fully connected network, one data point for training / testing was created for each site and frequency for the prediction of mean amplification. The number of datapoints per site equals the number of events at the site times the number of frequencies in the prediction of event-specific amplifications using the many-to-one NN layout.

The training and test sets for the prediction of event-specific amplifications using the many-to-one NN layout contain one data point for each frequency and each observation per site. The many-to-many design in the CNN requires just one data point per observation and site in the training and test sets.

NN Training and Prediction Results

We initially trained a network to predict event-specific site amplifications. As this turned out to be quite difficult and computationally expensive, we reverted to working with mean site

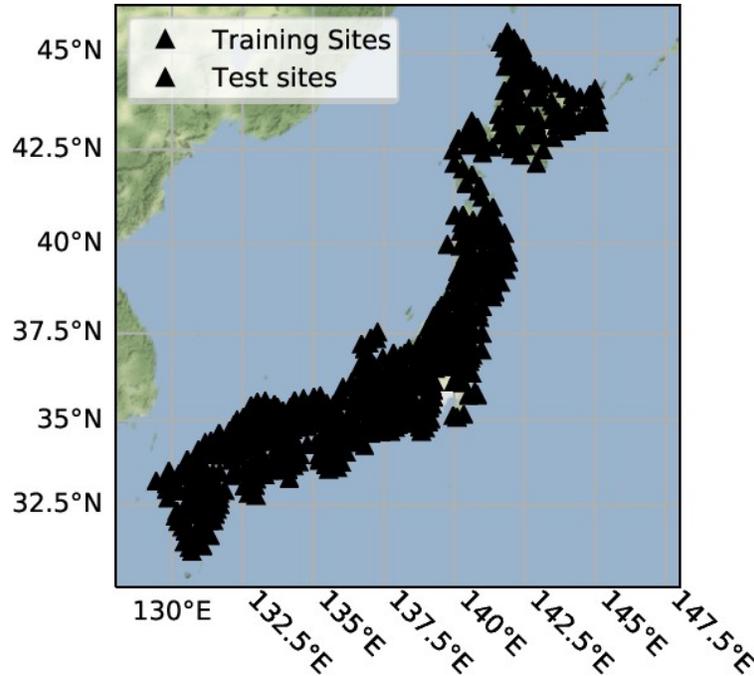


Figure 6: Locations of KiK-net sites assigned to training and test sets in this study.

amplification in a first step.

We trained the fully connected NN and convolutional NN with the mean AFs for 596 KiK-net sites assigned to the test set. Mini-batch gradient descent using the Adam optimizer [26] was carried out to minimize the mean square logarithmic error (MSLE) between observed and predicted theoretical AFs. We chose the MSLE as our loss function to incorporate the large range of amplifications observed between different sites and frequencies. A low MSLE is consistent with good visual agreements if AFs are plotted in logarithmic space as is conventionally done. We also report the mean absolute error (MAE) between predicted and observed amplification functions. The batch size was set to 2048 for the FCNN and to 50 for the CNN. We trained the ANNs for 1,000 epochs using the default learning rate of 10^{-3} .

Prediction of Mean Amplification Functions using the FCNN

Figure 7 shows the learning curve obtained during training of the fully connected neural network (FCNN) with mean amplification functions. The training loss (MSLE) is reduced from an initial value of 1.36 to 0.015 (Table 1). Input features (i.e., v_s and frequency of amplification) were standardized by removing the mean and scaling to unit variance using ‘StandardScaler’ from the scikit-learn library [27] before training. To control the amount of overfitting to the training

data, we used dropout regularization in the first five hidden layers. The dropout rate was adjusted to a value of 0.15 by trial and error. Lower values resulted in a higher validation error, while higher values increased the training error without further reducing the validation error. Figure 8

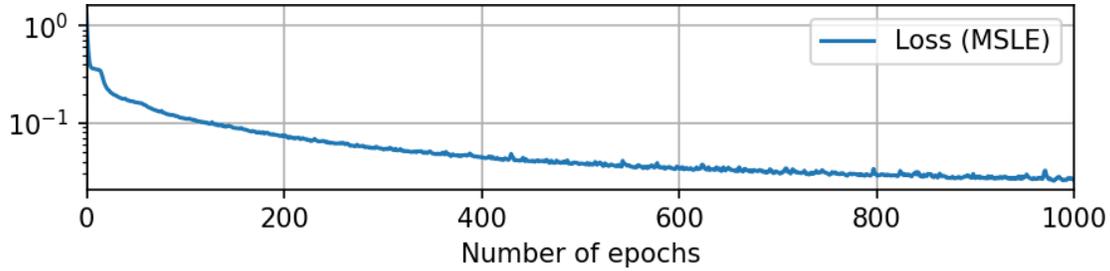


Figure 7: Learning curve with mean absolute training and validation errors during Adam optimization of the fully connected neural network (FCNN) for prediction of mean site response

	MSLE (train)	MAE (train)	MSLE (test)	MAE (test)
Baseline*			0.216	2.005
FC	0.015	0.521	0.140	1.593
CNN	0.021	0.595	0.104	1.307

Table 1: MAE and MSLE (loss) between observed amplifications and amplifications predicted from the preferred fully connected (FC) and convolutional (CNN) neural networks.

Theoretical SH1D amplification compares observed and predicted mean amplification functions for 9 randomly selected training sites. The low training error is reflected in the good match between observed and predicted mean amplification functions.

We used the trained FCNN to predict mean site amplifications at the 66 test sites and obtained a MSLE of 0.014 (Table 1). Figure 9 compares the observed and predicted mean amplifications at 9 randomly selected test sites. The MSLE at the displayed test sites ranges from about 0.040 for sites where the predicted AF is close to the observations (e.g., ISKH04, MIEH06, KOCH13) to values above 0.150 for sites where the predicted AF does not reproduce the observation (e.g., YMNH14, SRCH01, YMTH07) well. However, predicting the site response from a soil profile is generally difficult due to multi-dimensional effects, modeling inaccuracies and uncertainties in soil property estimates [e.g. 9, 10]. In order to put the quality of the AFs predicted by the NN into perspective, we compare them with a more conventional method for site response prediction. We computed theoretical SH1D AFs for a vertically incident plane wave, with densities and quality factors derived from the shear-wave velocity profiles using an empirical relation [28].

Theoretical AFs were smoothed in the same way as observed mean AFs. The MSLE between theoretical and observed AFs is listed for each site in Figure 9 (as well as Fig. 8 for reference, although we note that it makes little sense to compare training losses to theoretical predictions). With the exceptions of sites GIFH25 and SRCH01, the theoretical model results in a larger prediction error than the NN. The MSLE of 0.22 between theoretical and observed AFs

for the 66 test sites (Table 1) is used as a baseline to assess the NN’s performance, and shows that the NN is generally predicting mean AFs more accurately than the theoretical model.

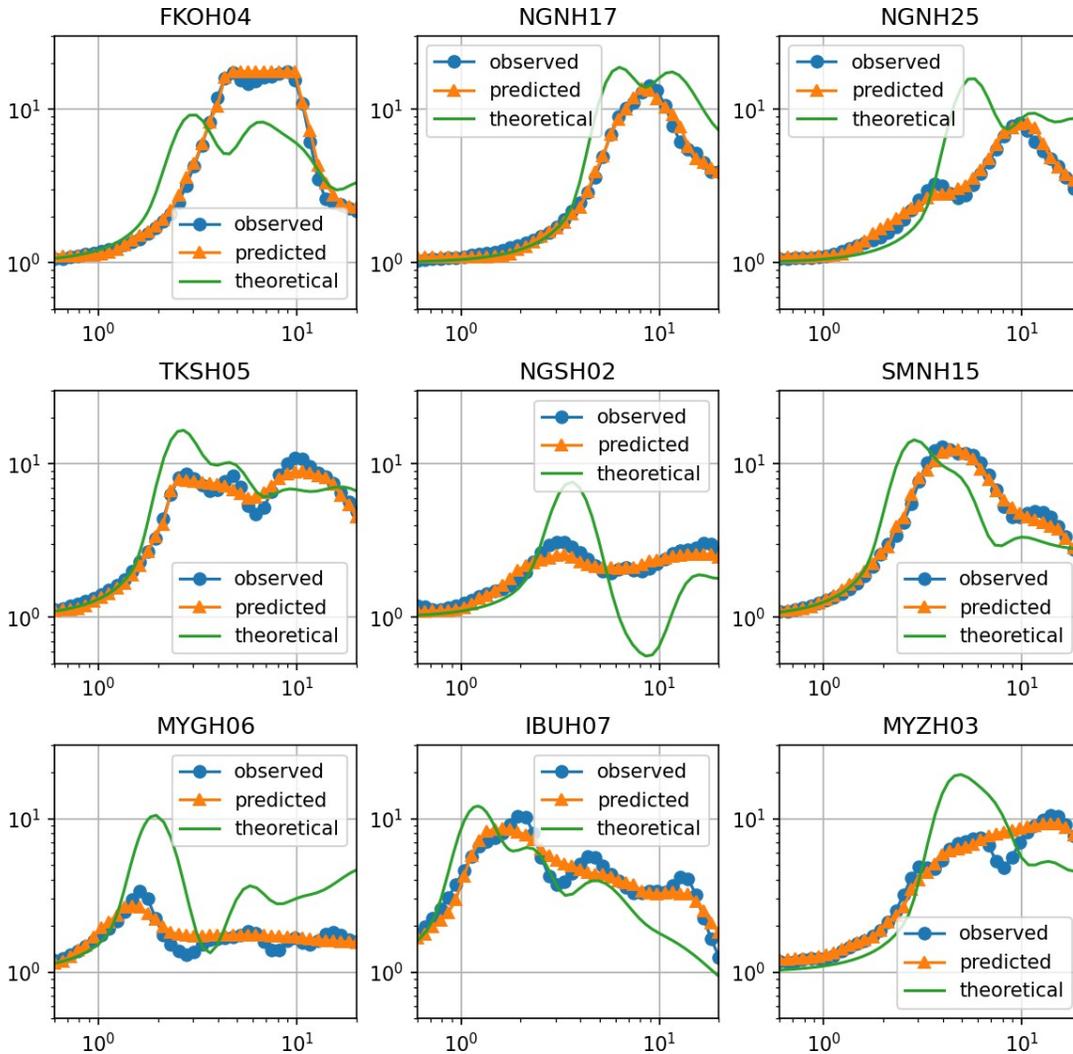


Figure 8: Comparison between observed mean amplification functions (blue) and amplification functions predicted by the FCNN (orange) for 9 randomly selected training sites. Solid green lines show theoretical 1D site amplification functions. Numbers in brackets next to the site name give the training loss (MSLE) for the site. Green numbers in the upper left corner show the baseline loss (based on the theoretical SH1D amplification function) for the site.

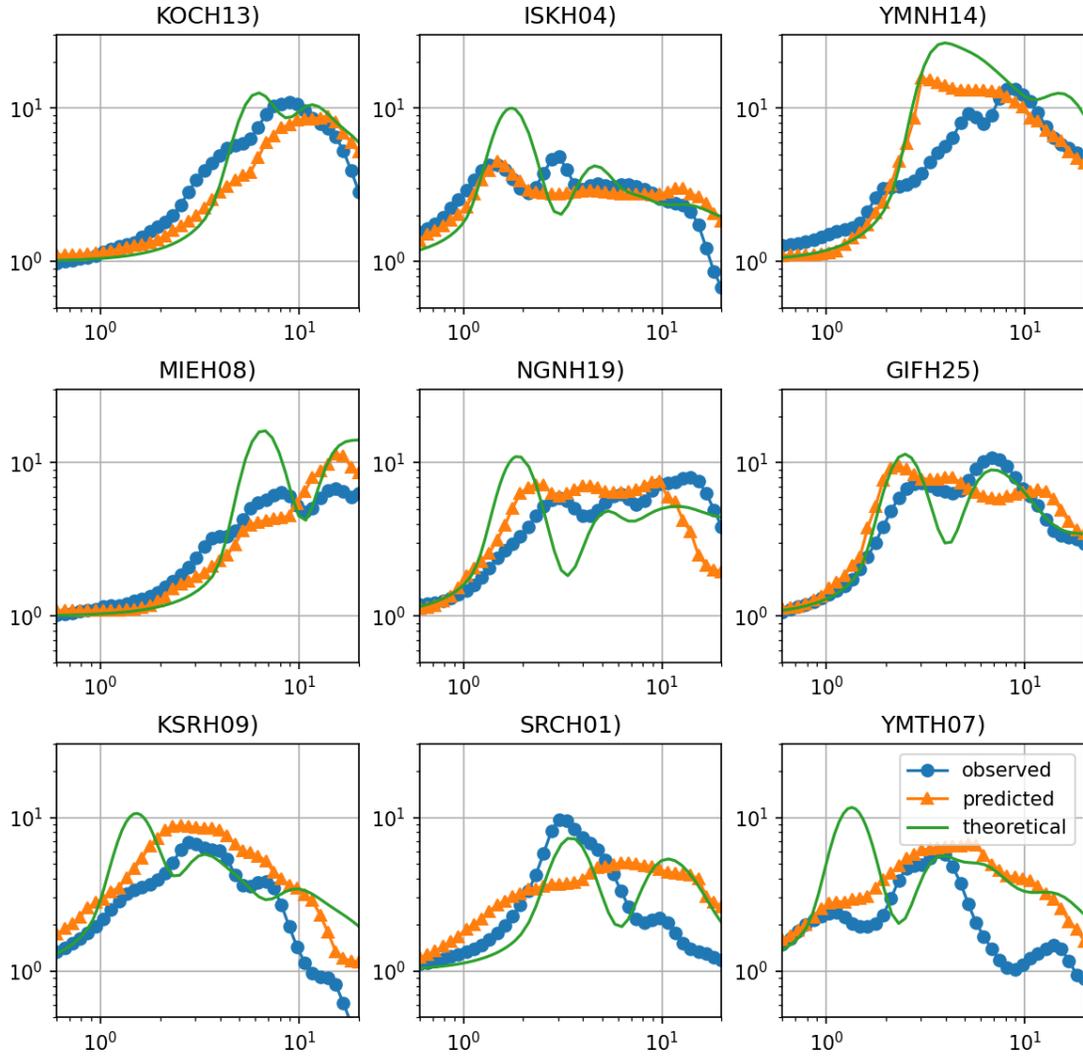


Figure 9: Same as Figure 8, but showing mean and predicted mean amplifications for 9 randomly selected test sites.

Prediction of mean amplification functions using a CNN

The convolutional CNN was trained using observed mean site amplifications for the same 596 training sites as for the FCNN. As with the FCNN, we adjusted drop-out rates for the CNN by trial and error to minimize the trade-off between high model bias (in case of poor performance for training sites) and high variance (in case of overfitting). For the CNN we tuned to drop-out rates to different values for each layer. A drop-out rate of 0.5 was used after the pooling layer (Fig. 2), while drop-out rates of 0.3, 0.15 and 0.10, respectively, were used for the three subsequent, fully connected hidden layers.

The CNN was trained for 2,000 epochs using a batch size of 50 sites. The loss was reduced from an initial value of 1.45 to a minimum of 0.043. We used the trained model to predict AFs for training and test sites and obtained a MSLE values of 0.021 and 0.104, respectively (Table 1). Note that the MSLE for predicted AFs at training sites (0.021, Table 1) is lower than the lowest minimum value obtained during optimization (0.043). This discrepancy is caused by drop-out regularization. Drop-out regularization randomly eliminates neurons during training, resulting in a relatively high training error. However, all neurons are enabled during prediction (drop-out rate is set to zero), which results in a lower prediction error than training error.

Both MSLEs and MAEs at test sites are lower for the CNN than for the FCNN. Moreover, the CNN achieves a test MSLE that is 50% lower than baseline (Table 1). Figure 10 compares observed, predicted and theoretical (baseline) AFs for the same 9 test sites as shown for the FCNN (Fig. 9). The CNN results in more accurate predictions especially for sites YMNH14 (MSLE reduction from 0.146 to 0.028), MIEH08 (0.043 to 0.020) and GIFH25 (0.064 to 0.030).

Figure 11a shows the distribution of the MSLE obtained by the baseline, FCNN and CNN at the 66 test sites. The CNN clearly represents an improvement with respect to both the FCNN and baseline. For example, the CNN distribution peaks at an MSLE of 0.05, with a median MSLE of 0.073. Baseline (SH1D) errors are more uniformly distributed with a median MSLE of 0.181. Compared to the FCNN, the CNN achieves an MSLE below 0.075 for more sites, and less sites with errors above 0.225. We also computed the change in MSLE between the two NN designs and the baseline for each site and plotted up the distribution of the change of error (Fig. 11b). With respect to baseline, the NNs results in an improvement at most sites; the CNN also outperforms the FCNN in terms of number of sites where an improvement is observed.

Summary and Outlook

We have calculated mean observed AFs for 662 KiK-net sites and 41 CSMIP geotechnical arrays. 95% of the sites were assigned as training sites, with the remaining 5% withheld as test sites. A FCNN and a CNN were trained to predict the observed amplification functions from a discretized representation of the velocity profiles.

Both NN designs converged to a solution with minimal loss, and accurately reproduced the observed AFs at the training sites. While the quality of the prediction at the test sites varied, both the FCNN and the CNN outperformed predictions based on the theoretical SH1D site

response in terms of MSLE (mean squared logarithmic error) between observed and predicted AFs.

Predictions made by the CNN resulted in an MSLE that was 50% lower than the SH1D baseline, and 25% lower than the predictions by the FCNN. Proper regularization and fine fine-tuning of the drop-out rate was found to be essential to obtain good predictions at test sites not used for training.

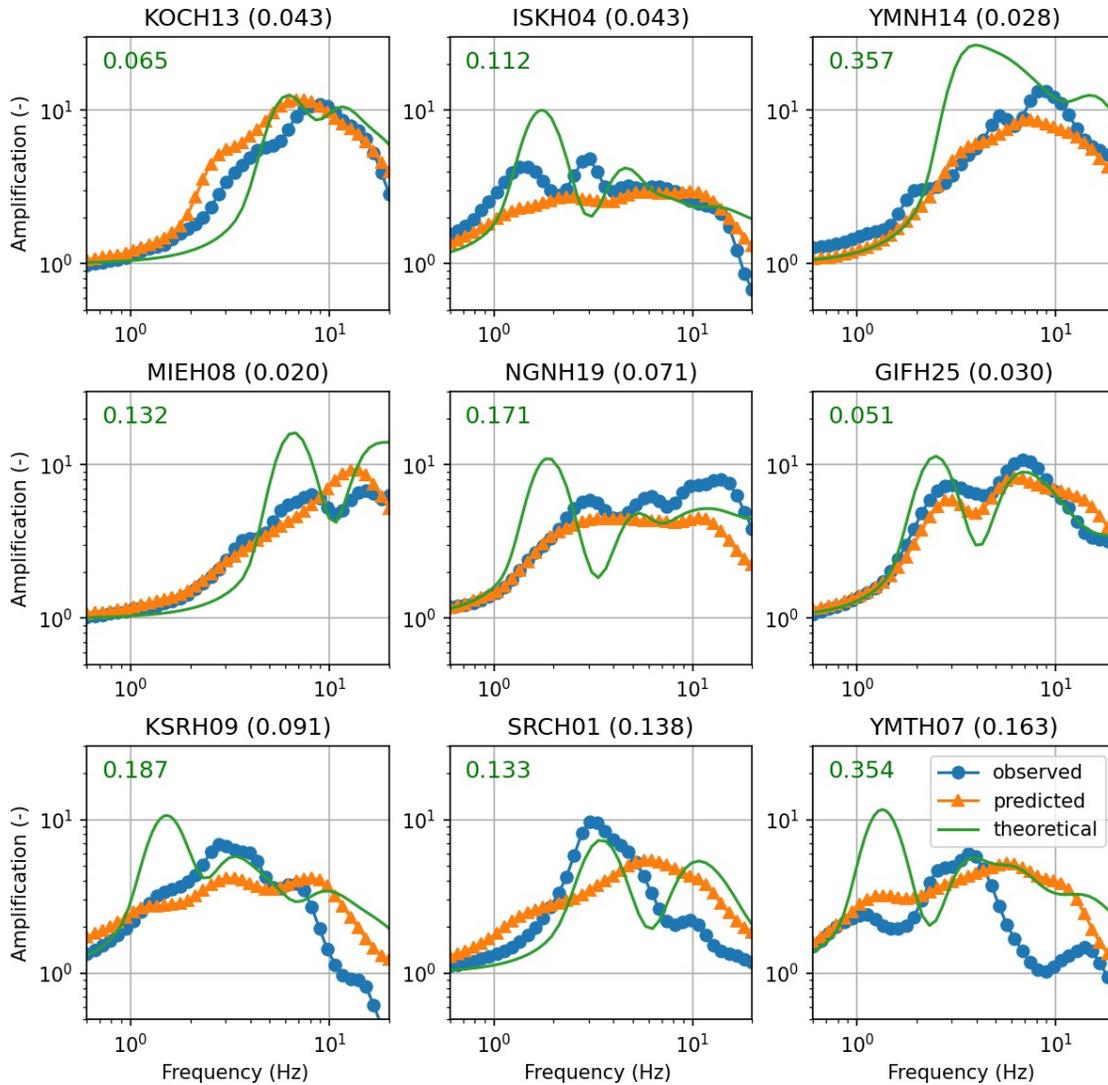


Figure 10: Same as Figure 9, but showing predictions by the convolutional neural network (CNN) at the same test sites.

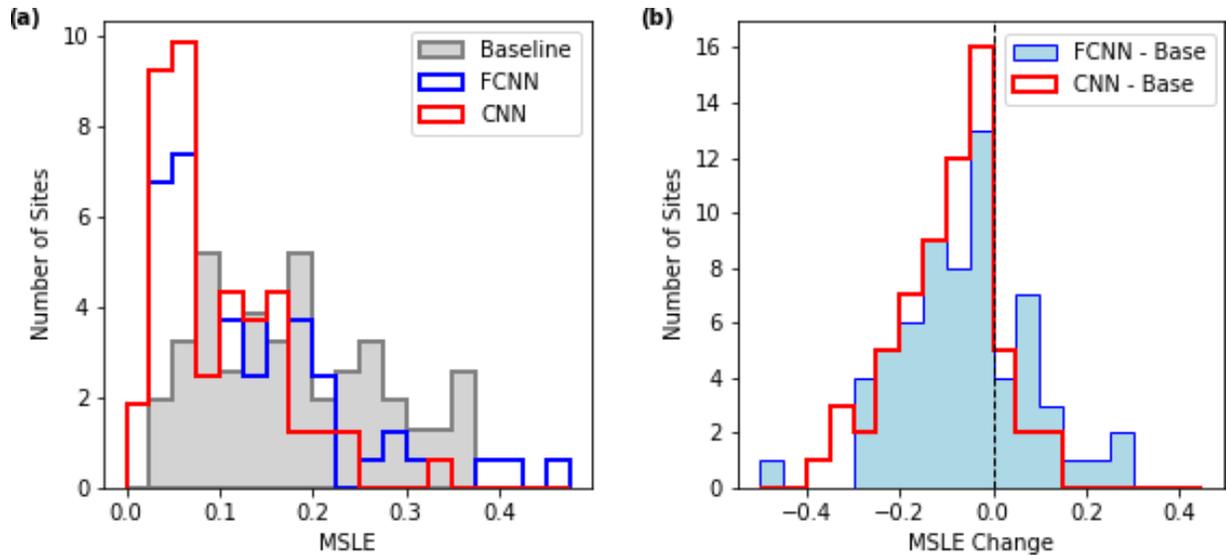


Figure 11: (a) Distribution of site-specific MSLE obtained from baseline, FCNN and CNN results. (b) Distribution of change in MSLE achieved by use of FCNN and CNN with respect to the baseline ('Base').

These results show that artificial NNs have the potential to take advantage of the full velocity profile information for more accurate predictions of observed AFs. Although a simple SH1D amplification function does not represent the state of the art for site-specific seismic hazard analysis in engineering seismology, it serves as a useful benchmark that demonstrates the limits of commonly made assumptions (in particular a horizontal, 1D layered structure and a vertically incident plane wave). A NN which learns to predict AFs purely from data is not bound by such assumptions, and we have demonstrated the level of improved accuracy with respect to the baseline that can be learned from data.

In future work, we will also evaluate the accuracy of the NN-predicted AFs against empirical site amplification functions, which are typically based on the V_{S30} . While our efforts in this paper were focused on the prediction of mean AFs, future work should address the prediction event-specific amplification. Here, effects of incident wavefield characteristics, scattering and nonlinearity would be captured by feeding the network with earthquake magnitude, hypocentral distance, and input signal metrics (e.g. PGA, spectral accelerations at different frequencies, duration). In case of a CNN, this requires a slightly more complicated design with mixed data inputs. We also recommend for future work to explore the use of information on the multi-dimensionality of a site structure.

The best-performing network design will be re-trained using all the extracted AFs and made available to the community for data-informed site response prediction using the full soil profile.

Disclaimer

The contents of this report were developed under Contract No. 1019-014 from the California Department of Conservation, California Geological Survey, Strong Motion Instrumentation Program. However, these contents do not necessarily represent the policy of that agency or endorsement by the State Government.

Acknowledgements

This project is supported by the California Department of Conservation, California Geological Survey, Strong Motion Instrumentation Program, Contract 1019-014. The authors thank Daniel Swensen at the California Department of Conservation for providing soil profiles of CSMIP vertical array sites. We are grateful to the Center for Engineering Strong Motion Data (CESMD) and the National Research Institute for Earth Science and Disaster Resilience (NIED) for providing vertical array strong motion data. Data pre-processing was carried out on the OLCF (Oak Ridge Leadership Computational Facility) cluster Rhea under an INCITE research allocation.

References

- [1] L. Bonilla, C. Gelis, and J. Regnier. The challenge of nonlinear site response: field data observations and numerical simulations. In *4th IASPEI/IAEE International Symposium: Effects of Surface Geology on Seismic Motion, August 23-26, 2011, University of Santa Barbara*, 2011.
- [2] E. M. Thompson, Y. Tanaka, L. G. Baise, and R. E. Kayen. Three-dimensional site response at KiK-net downhole arrays. In *7th International Conference on Urban Earthquake Engineering & 5th International Conference on Earthquake Engineering*, pages 3–5, 2010.
- [3] D. Roten, D. Fäh, and L. Bonilla. High-frequency ground motion amplification during the 2011 Tohoku earthquake explained by soil dilatancy. *Geophys. J. Int.*, in press, 2013.
- [4] D. Roten, D. Fäh, and L. Bonilla. Quantification of cyclic mobility parameters in liquefiable soils from inversion of vertical array records. *Bull. seism. Soc. Am.*, 104(6):3115–3138, 2014.
- [5] J. Douglas. Earthquake ground motion estimation using strong-motion records: a review of equations for the estimation of peak ground acceleration and response spectral ordinates. *Earth-Science Reviews*, 61(1-2):43–104, 2003.
- [6] F. O. Strasser, N. A. Abrahamson, and J. Bommer. Sigma: Issues, insights, and challenges. *Seismological Research Letters*, 80(1):40–56, 2009.
- [7] J. Bommer and N. Abrahamson. Why do modern probabilistic seismic-hazard analyses often lead to increased hazard estimates? *Bull. seism. Soc. Am.*, 96(6):1967–1977, 2006.

- [8] N. A. Abrahamson, W. J. Silva, and R. Kamai. Summary of the ASK14 ground motion relation for active crustal regions. *Earthquake Spectra*, 30(3):1025–1055, 2014.
- [9] E. M. Thompson, L. G. Baise, R. E. Kayen, and B. B. Guzina. Impediments to predicting site response: Seismic property estimation and modeling simplifications. *Bulletin of the Seismological Society of America*, 99(5):2927–2949, 2009.
- [10] E. M. Thompson, L. G. Baise, Y. Tanaka, and R. E. Kayen. A taxonomy of site response complexity. *Soil Dynamics and Earthquake Engineering*, 41:32–43, 2012.
- [11] F. Gatti, F. Lopez-Caballero, D. Clouteau, and R. Paolucci. On the effect of the 3-D regional geology on the seismic design of critical structures: the case of the Kashiwazaki-Kariwa Nuclear Power Plant. *Geophysical Journal International*, 213(2):1073–1092, 2018.
- [12] Z. Hu, D. Roten, O. K.B., and S. Day. Modeling of empirical transfer functions with 3d velocity structure. *submitted to Bull. Seism. Soc. Am.*, 2020.
- [13] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.
- [14] G. Nolet. *A breviary of seismic tomography*. Cambridge University Press, 2008.
- [15] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
- [16] P. M. DeVries, T. B. Thompson, and B. J. Meade. Enabling large-scale viscoelastic calculations via neural network acceleration. *Geophysical Research Letters*, 44(6): 2662–2669, 2017.
- [17] F. Chollet et al. Keras: the Python deep learning API, 2015. <https://keras.io/>, [Last accessed October 2020].
- [18] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from [tensorflow.org/](https://www.tensorflow.org/), [Last accessed October 2020].
- [19] CESMD. Center for Engineering Strong Motion Data, 2020. <https://www.strongmotioncenter.org/>, [Last accessed October 2020].
- [20] A. Shakal and M. Huang. Standard tape format for CSMIP strong-motion data tapes. Report osms 85-03, California Department of Conservation, Division of Mines and

- Geology, 1985.
- [21] K. Konno and T. Ohmachi. Ground-motion characteristics estimated from spectral ratio between horizontal and vertical components of microtremor. *Bull. seism. Soc. Am.*, 88(1): 228–241, 1998.
- [22] K. Reitz. Requests: HTTP for Humans™, 2020. <https://requests.readthedocs.io/en/master/>, [Last accessed October 2020].
- [23] NEID. Strong-motion seismograph networks (k-net, kik-net), 2020. <https://www.kyoshin.bosai.go.jp/>, [Last accessed October 2020].
- [24] O. D. Team. A Python Framework for Seismology, 2020. <https://docs.obspy.org/>, [Last accessed October 2020].
- [25] T. A. S. Foundation. Apache Spark™ - Unified Analytics Engine for Big Data, 2020. <https://spark.apache.org/>, [Last accessed October 2020].
- [26] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint; arXiv:1412.6980*, 2014.
- [27] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. scikit-learn.org/, [Last accessed October 2020].
- [28] T. Brocher. Key elements of regional seismic velocity models for ground motion simulations. In *Proc. of Int. Workshop on Long-Period Ground Motion simulations and velocity structures, Tokyo, Nov 14-15, 2006*, 2006.