Short Manual for training an ANN

1. ANN architecture

The new ANN architecture consists of four input branches. The first two correspond to the long-period spectral accelerations of the two horizontal components. The third branch includes source parameters, while the fourth represents the EC8 site classification (categories A, B, C, or D), encoded in binary form. Each input branch is first processed independently through fully connected layers, after which they are merged via a concatenation layer. The merged representation is then passed to a fully connected hidden layer with 40 neurons. Finally, this shared hidden layer connects to two separate output layers, each predicting the short-period spectral accelerations of one horizontal component. For the vertical component the second branch is deactivated and the number of neurons of the common hidden layer is reduced to 20. To ensure comparability across variables, z-score normalization is applied to the first three input branches. Hidden layers employ hyperbolic tangent activation functions, whereas the output layers use linear activation functions, with the number of nodes matching the dimensionality of the target values. The architecture of the ANN is defined inside the code train_ann_basics_h12.m

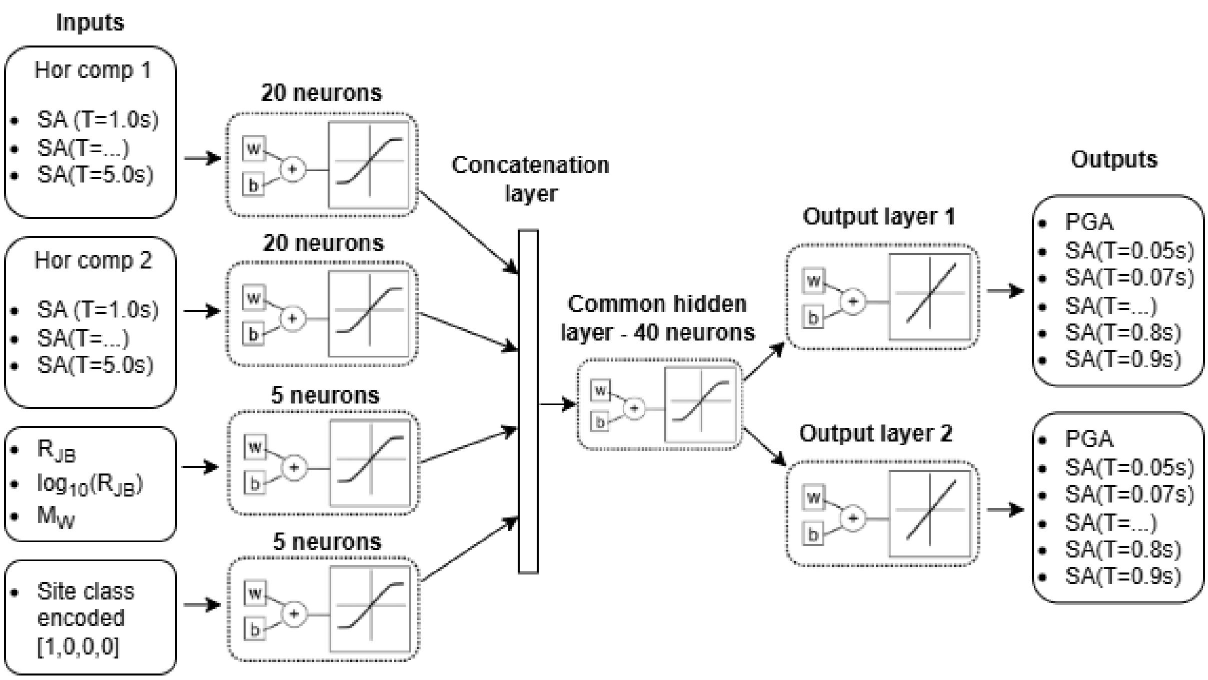

Figure 1. Architecture of the ANN.

2. Training

For training the ANNs you have to run the code drive.m. Inside you have the change the following options:

- dbn_name = 'ESM_SIMBADs'; Name of the database with the spectral accelerations for training. It should be located inside the subfolder database.
- TransferLearning = 'False'; Use 'True' if you want to do a two-step training with transfer learning. In that case you need to define also dbn_name2, which is the database for the second training (fine-tunning), otherwise dbn_name2 is ignored.
- dbn_name2 = 'ESM_SIMBADs'; If TransferLearning = 'False' just use the same name as in dbn_name.
- num_nets = 10; Number of individual nets to be trained.
- n_LoopsANN = 1; Number of trained nets before choosing the best one. (1 or 2).
- add_distance = 'True'; 'True' if you want to add R_{JB} as input for training.
- add_m = 'True'; 'True' if you want to add M_W as input for training.
- add_lndistance ='True'; 'True' if you want to add log10(R_{JB}) as input for training.
- separate_classes = 'False'; 'True' for activating the fourth branch of the ANN, i.e., EC8 site classes.
- ann.trn.nr = 2; Choose 2 for training ANNs for horizontal (h12) and vertical (ud) components at the same time.
- TnC = [1,1]; Corner periods T* of the ANNs.
- cp = {'h12','ud'}; Components (ud=vertical;h12=both horizontals).

The trained nets will be saved in the subfolder ANNs. Their names will have the following format: net_(100*Tnc)_ALL_(cp)_(input variables)_(dbn_name)_v(##). For instance, for the set options above, the names would be:

- Horizontal components:net_100_ALL_h12_Rjb_Mw_logRjb_ESM_SIMBADs_v##
- Vertical component:net_100_ALL_ud_Rjb_Mw_logRjb_ESM_SIMBADs_v##

Ten already trained nets for the horizontal and vertical components are included in subfolder ANNs.

3. Database

The database(s) for training should be inside the subfolder database. As a minimum it should contain spectral accelerations (PSA) for the three components computed at these 30 periods: [0,0.04,0.05,0.07,(0.1:0.05:0.5),0.6,0.7,0.75,0.8,0.9,1.0:0.2:2.0,(2.5:0.5:5)], and Mw, site class and R_{JB} if they are going to be used. The units of PSA should be in cm/s^2.The 30 PSA are internally divided into input and output depending on T* (TnC).

A combination of SIMBAD and the ESM databases is included (ESM_SIMBADs.mat), consisting of 2497 strong-motion records, with magnitudes M_W from 5 to 7.8, and R_{JB} lower than 110 km. Figure 2 shows the magnitude vs distance distribution of the records.

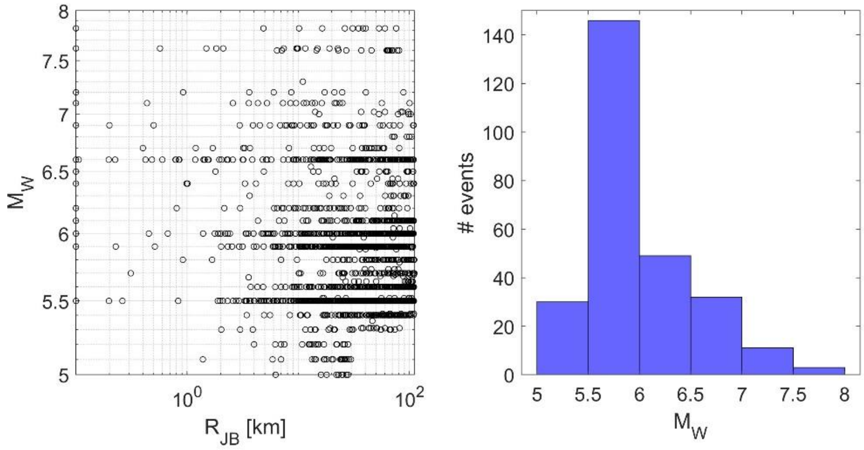

Figure 2. Magnitude vs distance R_{JB} scatter plot of recordings in the ESM+SIMBAD data set used for training the ANN (left). Histogram of the number of events per magnitude bin (right).

The format of the database should be as follows (a Matlab structure named SIMBAD), with the same variable names:

1x2497 struct with 36 fields

Fields	rec	event_name					Mw						site_EC8
1	1	'AM-1988-00...	...	...	...	...	6.70...	''	'...	'...	...	...	'C'
2	2	'AM-1990-00...	...	...	...	...	5.50...	''	'...	'...	...	...	'B'
3	3	'AM-1990-00...	...	...	...	...	5.50...	''	'...	'...	...	...	'A'
4	4	'AM-1990-00...	...	...	...	...	5.50...	''	'...	'...	...	...	'B'
5	5	'AM-1990-00...	...	...	...	...	5.50...	''	'...	'...	...	...	'C'
6	6	'EMSC-2004...	...	...	...	...	5.50...	''	'...	'...	...	...	'A'
7	7	'EMSC-2004...	...	...	...	...	5.50...	''	'...	'...	...	...	'A'
8	8	'EMSC-2004...	...	...	...	...	5.50...	''	'...	'...	...	...	'E'
9	9	'EMSC-2004...	...	...	...	...	5.50...	''	'...	'...	...	...	'B'
10	10	'EMSC-2007...	...	...	...	...	5.90...	''	'...	'...	...	...	'B'
11	11	'EMSC-2007...	...	-...	-...	...	5.50...	''	'...	'...	-...	-...	'A'
12	12	'EMSC-2008...	...	-...	-...	...	5.70...	''	'...	'...	-...	-...	'A'
13	13	'EMSC-2008...	...	-...	...	67	5.70...	''	'...	'...	-...	...	'B'
14	14	'EMSC-2008...	...	-...	...	81	5.60...	''	'...	'...	-...	...	'C'

Fields					Rjb												psa_h1	psa_h2	psa_v
1	...	...	...	...	30.8735	...	'...	'...	'...	'...	'...	'A'	[...	[...	[...	[...	*37x1 dou...*	*37x1 dou...*	*37x1 dou...*
2	...	...	...	...	5.9553	...	'...	'...	'...	'...	'...	'A'	[...	[...	[...	[...	*37x1 dou...*	*37x1 dou...*	*37x1 dou...*
3	...	...	...	...	14.5794	...	'...	'...	'...	'...	'...	'A'	[...	[...	[...	[...	*37x1 dou...*	*37x1 dou...*	*37x1 dou...*
4	...	...	...	...	25.8610	...	'...	'...	'...	'...	'...	'A'	[...	[...	[...	[...	*37x1 dou...*	*37x1 dou...*	*37x1 dou...*
5	...	...	...	...	46.1043	...	'...	'...	'...	'...	'...	'A'	[...	[...	[...	[...	*37x1 dou...*	*37x1 dou...*	*37x1 dou...*
6	...	...	...	...	102.8533	...	'...	'...	'...	'...	'...	'...	[...	[...	[...	[...	*37x1 dou...*	*37x1 dou...*	*37x1 dou...*
7	...	...	...	...	80.0241	...	'...	'...	'...	'...	'...	'...	[...	[...	[...	[...	*37x1 dou...*	*37x1 dou...*	*37x1 dou...*
8	...	...	...	...	82.7041	...	'...	'...	'...	'...	'...	'...	[...	[...	[...	[...	*37x1 dou...*	*37x1 dou...*	*37x1 dou...*
9	...	...	...	...	98.6419	...	'...	'...	'...	'...	'...	'...	[...	[...	[...	[...	*37x1 dou...*	*37x1 dou...*	*37x1 dou...*
10	...	...	...	...	103.1640	...	'...	'...	'...	'...	'...	'...	[...	[...	[...	[...	*37x1 dou...*	*37x1 dou...*	*37x1 dou...*
11	...	...	...	...	44.8743	...	'...	'...	'...	'...	'...	'...	[...	[...	[...	[...	*37x1 dou...*	*37x1 dou...*	*37x1 dou...*
12	...	...	...	...	48.5003	...	'...	'...	'...	'...	'...	'...	[...	[...	[...	[...	*37x1 dou...*	*37x1 dou...*	*37x1 dou...*
13	...	...	...	...	25.6043	...	'...	'...	'...	'...	'...	'...	[...	[...	[...	[...	*37x1 dou...*	*37x1 dou...*	*37x1 dou...*
14	...	...	...	...	102.0132	...	'...	'...	'...	'...	'...	'...	[...	[...	[...	[...	*37x1 dou...*	*37x1 dou...*	*37x1 dou...*

Figure 3. Format of the Matlab structure with the data used for training.

Paolucci, R., Gatti, F., Infantino, M., Smerzini, C., Ozcebe, A. G., and Stupazzini, M. (2018). Broadband ground motions from 3D physics-based numerical simulations using artificial neural networks. Bull. Seismol. Soc. Am., 108(3):1272–1286.

Hernández-Aguirre, V. M., Rupakhety, Paolucci, R., & Smerzini, C. (2025). Physics-based Broadband Ground Motion Simulations Enhanced by Transfer Learning. *Manuscript in Preparation*.