Train ANNs in MATLAB

Siyu Qi

June 3, 2020

1 Introduction

In this step-to-step tutorial, we will learn how to train and test ANNs in MATLAB. We will be training two models with different architectures: single-output or multi-output dense neural network.

2 Contents

1. Preparation

2. Train ANNs

3. Test ANNs

3 Preparation

3.1 Prerequisites

This code has been tested with MATLAB 2019a and requires MATLAB Deep Learning Toolbox to run. To check your software version and print a list of installed toolboxes, simply type ver in the command window and you will see outputs like below.

Listing 1: Print MATLAB version and toolboxes

```
 1  >> ver
 2  _____
 3  MATLAB Version: 9.6.0.1150989 (R2019a) Update 4
 4  MATLAB License Number: 589250
 5  Operating System: Mac OS X  Version: 10.15.4 Build: 19E287
 6  Java Version: Java 1.8.0_181-b13 with Oracle Corporation Java HotSpot(TM)
       64-Bit Server VM mixed mode
 7  _____
 8  MATLAB                          Version 9.6         (R2019a)
 9  Simulink                        Version 9.3         (R2019a)
10  Control System Toolbox          Version 10.6        (R2019a)
11  DSP System Toolbox              Version 9.8         (R2019a)
12  Deep Learning Toolbox           Version 12.1        (R2019a)
13  Image Processing Toolbox        Version 10.4        (R2019a)
14  ...
```

If Deep Learning Toolbox shows up in the list, you are good to skip the rest of this section and go to 3.2. If not, go to the Home tab in MATLAB and, in the Environment section, click the Add-Ons icon.
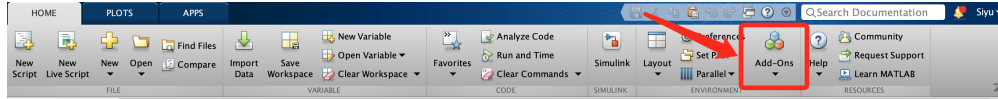
Figure 1: Find Deep Learning Toolbox

As Add-On Explorer opens, type "Deep Learning" in the search field. Click the first searching result and you shall be able download the toolbox on that page.
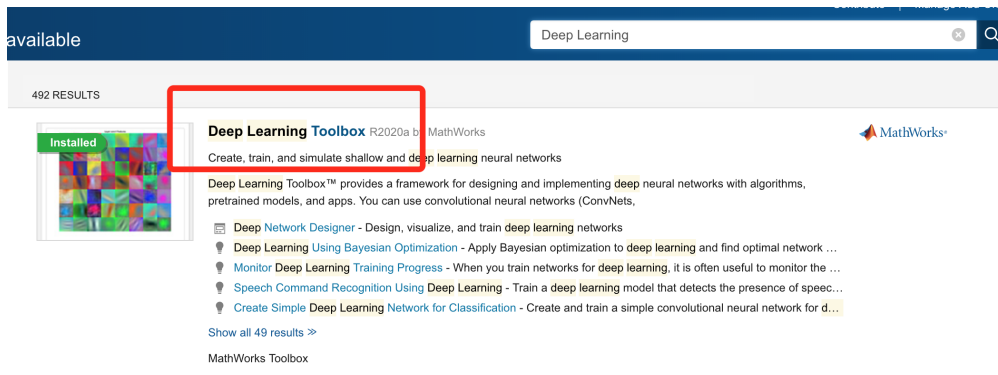

Figure 2: Download Deep Learning Toolbox

3.2 Download Code

Go to the Github repository `https://github.com/QiSiyu/CalSim-ANN` and click on the green button `Clone or download` then click `Download ZIP`. Unzip the folder on your computer.
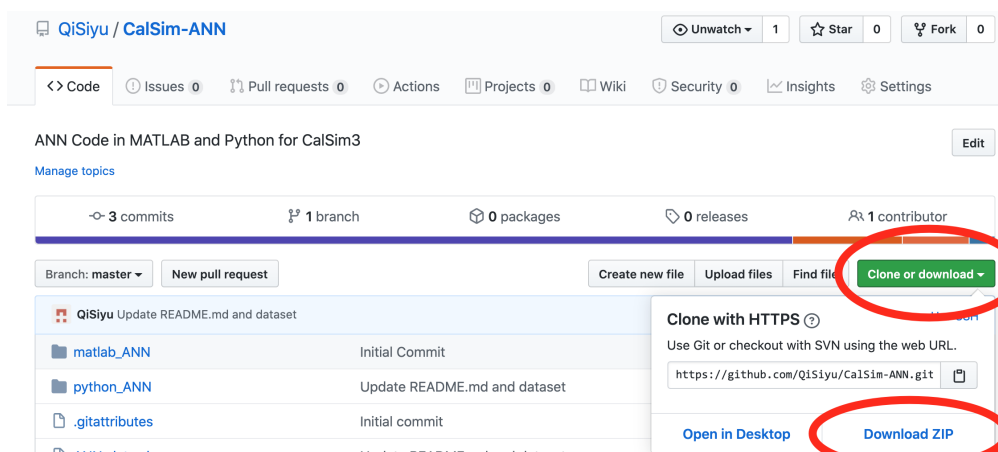

Figure 3: Download code from Github

We will use the folder `matlab_ANN` and the excel file `ANN_data.xlsx` in this tutorial. Here is a list of files in folder "matlab_ANN":

1. `trainANNs_single_output.m` and `trainANN_multi_output.m` : scripts to train single- or multi-output ANNs.

2. testANNs_single_output.m and testANNs_multi_output.m : scripts to test trained ANNs.

3. network : folder containing trained ANN models.

4. utils : folder containing helper functions.

Now we are ready to train ANNs on our own computer.

4 Train ANNs

4.1 Train a Single-Output ANN

Open trainANNs_single_output.m in "matlab_ANN" folder. Find the "User Settings" section in line 25 of the script.

Listing 2: User Settings in training scripts

```
1  % **********************************************************
2  % ********************* User Settings **********************
3  % **********************************************************
4
5  % 1. Define locations to be predicted:
6  % available:'Emmaton','Jersey Point','Collinsville', 'Rock Slough',
7  % 'Antioch', 'Mallard', 'LosVaqueros', 'Martinez', 'MiddleRiver', 'Vict
8  % Intake', 'CVP Intake', 'CCFB_OldR'
9  predict_stations = {'Emmaton'};%,'Jersey Point',...
10 %                    'Collinsville', 'Rock Slough',...
11 %                     'Antioch','Mallard','LosVaqueros',...
12 %                     'Martinez','MiddleRiver','Vict Intake',...
13 %                     'CVP Intake','CCFB_OldR'};
14
15 available_inputs = {'SAC','Exp','SJR','DICU','Vern','SF_Tide','DXC'}; % do
       not change
16 % 2. Define variables to be used for prediction:
17 input_var = {'SAC','Exp','SJR','DICU','Vern','SF_Tide','DXC'};
18
19 % 3. Define directory to the input and output excel file:
20 % note: no blank space is allowed in DATA_DIR or FILE_NAME
21 DATA_DIR = '/Users/siyuqi/Downloads';
22 FILE_NAME = 'ANN_data.xlsx';
23
24 % 4. Define name of folder you want to save your ANN
25 ANNsetting ='single_output_ANN-0.1-0.9-8-2-1-80%-MEM-7-10-11'; % folder to
       put results in
26
27 % 5. (optional) Modify num of neurons and activation func in hidden layers
```

```
28  % Notes: current setting is [8, 2],
29  % this code only works for ANNs with 2 hidden layers
30  layers = {[8 2]};
31  layerTypes = {{'logsig','logsig','purelin'}};
32
33  % ************************************************************
34  % **************** User Settings Finished *****************
35  % ************************************************************
```

Users may change these 5 variables as needed:

1. predict_stations : a list of **one or more** stations that we want to train an ANN for.

2. input_var : a list of **one or more** variables to be used as input.

3. DATA_DIR and FILE_NAME : path to the excel file and the name of excel file, respectively.

4. ANNsetting : the name of folder to save the trained model.

5. layers and layerTypes : number of neurons in hidden layers and the activation functions of each layer. No need to modify them if user wants to use the default architecture.

After modifying the user settings, hit "Run" button to train the models. Again, one model will be trained for each station defined in predict_stations .

Trained models and generated fortran file will appear in a folder named as ANNsetting defines, e.g. single_output_ANN-0.1-0.9-8-2-1-80%-MEM-7-10-11 , inside a folder called network .
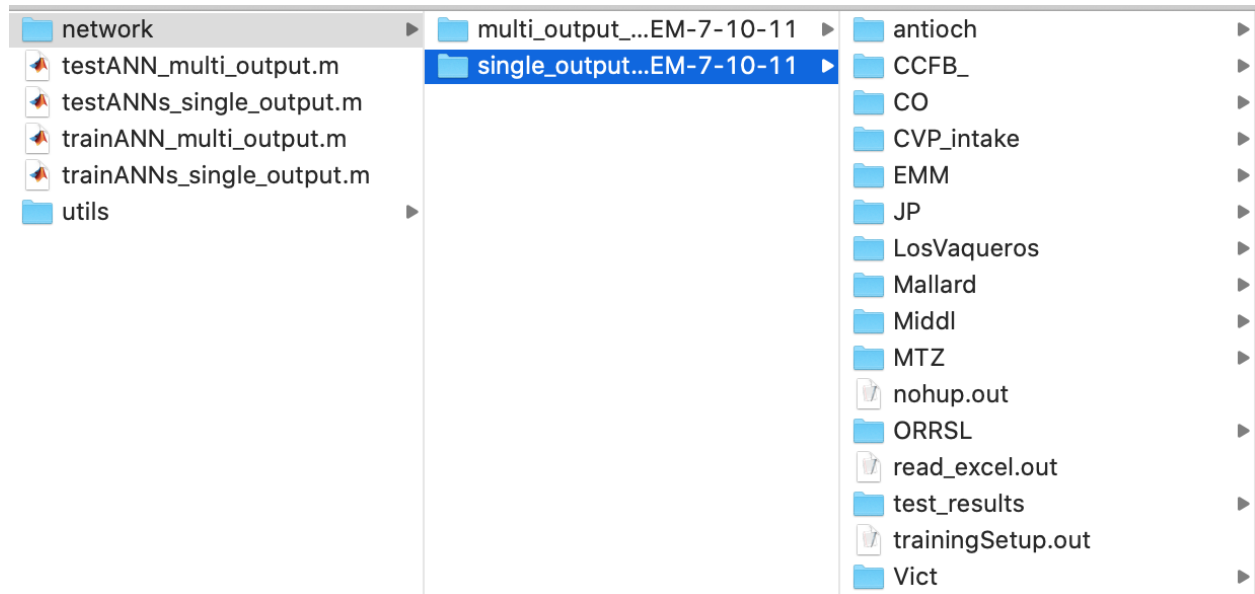

Figure 4: Output files for training single-output ANNs

V

4.2   Train a Multi-Output ANN

To train an integrated ANN, simply open trainANNs_multi_output.m in "matlab_ANN" folder. Then follow the instruction in section 4.1. Note that this script trains **one** ANN for all the stations defined in predict_stations .
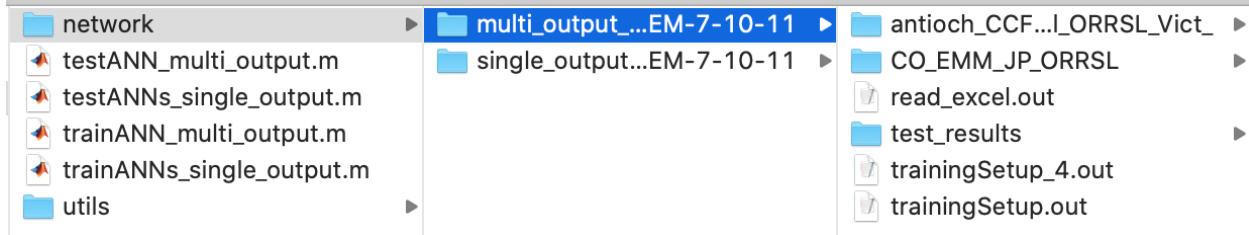

Figure 5: Output files for training multi-output ANNs

5   Test ANNs

To test the ANN, we load the trained model, pass inputs to the model and record the outputs in text files.

5.1   Test a Single-Output ANN

Open testANNs_single_output.m in "matlab_ANN" folder. Find the "User Settings" section in line 16 of the script.

Listing 3: User Settings in test scripts

```
1  % **********************************************************
2  % ********************* User Settings *********************
3  % **********************************************************
4
5  % 1. Define locations to be predicted:
6  % available:'Emmaton','Jersey Point','Collinsville', 'Rock Slough',
7  % 'Antioch', 'Mallard', 'LosVaqueros', 'Martinez', 'MiddleRiver', 'Vict
8  % Intake', 'CVP Intake', 'CCFB_OldR'
9  predict_stations = {'Emmaton','Jersey Point',...
10 'Collinsville', 'Rock Slough',...
11 'Antioch','Mallard','LosVaqueros',...
12 'Martinez','MiddleRiver','Vict Intake',...
13 'CVP Intake','CCFB_OldR'};
14
15 available_inputs = {'SAC','Exp','SJR','DICU','Vern','SF_Tide','DXC'}; % do
       not change
16 % 2. Define variables to be used for prediction:
17 input_var = {'SAC','Exp','SJR','DICU','Vern','SF_Tide','DXC'};
18
```

```
19  % 3. Define directory to the input and output excel file
20  % Note: no blank space is allowed in DATA_DIR or FILE_NAME
21  DATA_DIR = '/Users/siyuqi/Downloads/';
22  % DATA_DIR = 'D:/ANN/MATLAB/Data';
23  FILE_NAME = 'ANN_data2.xlsx';
24
25  % 4. Define ANNsetting (the folder where the model is saved). Must be same
26  % as training:
27  ANNsetting ='single_output_ANN-0.1-0.9-8-2-1-80%-MEM-7-10-11';
28
29  % 5. define whether to normalize outputs or not.
30  % Note: if set to true, output ec values are normalized between lowScale
31  % and highScale.
32  normalize_ec = true;
33
34  % 6. (optional) Define saving data precision (number of digits after the
35  % decimal point)
36  save_precision = 6;
37
38  % 7. (optional) set test_number to number of days users want to examine. If
39  % set to inf or negative, all the available inputs are sent to ANN.
40  test_size = inf;
41
42
43  % ************************************************************
44  % *************** User Settings Finished *****************
45  % ************************************************************
```

Users may change these 5 variables as needed.

1. predict_stations : a list of **one or more** stations for which we want to test the trained model.

2. input_var : a list of **one or more** variables to be used as input, must be same as in training script.

3. DATA_DIR and FILE_NAME : path to the excel file and the name of excel file, respectively.

4. ANNsetting : the name of folder to save the trained model, must be same as in training script.

5. normalize_ec : if true, record normalized (0.1 to 0.9); if false, record denormalized ec values.

6. save_precision : number of digits after the decimal point when recording model outputs.

7. test_size : number of test samples (in case user wants to test with a subset of the dataset). Set to 'inf' to use all the samples in dataset.

After modifying the user settings, hit "Run" button to test the model(s).

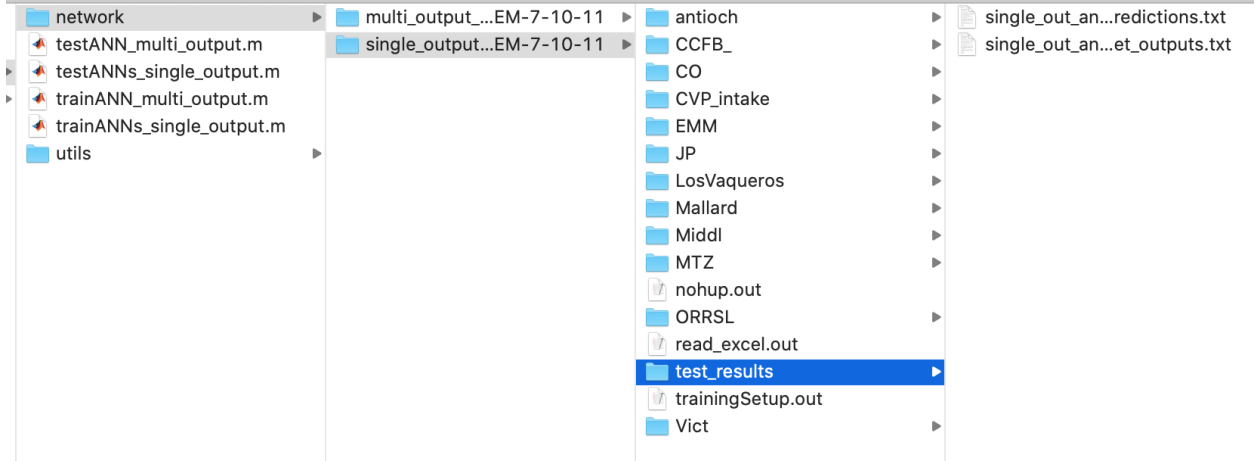Test results will appear in a folder named as ANNsetting defines, inside a folder called network .

Figure 6: Output files for testing single-output ANNs

5.2   Test a Multi-Output ANN

To test an integrated ANN, simply open testANN_multi_output.m in "matlab_ANN" folder. Then follow the instruction in section 5.1. Note that this script tests **one** ANN for all the stations defined in predict_stations .
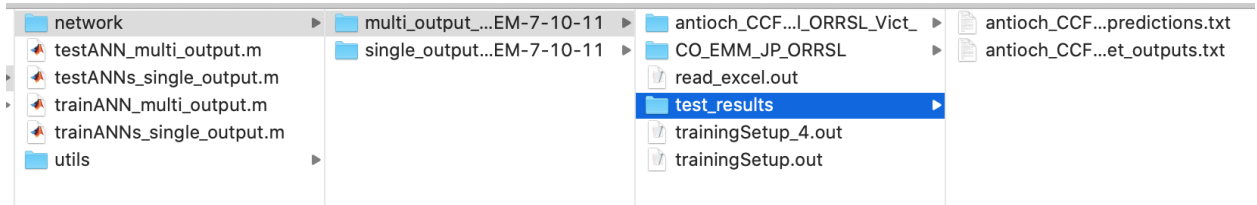

Figure 7: Output files for testing multi-output ANNs

VIII