**NEURAL NETWORKS PROJECT 3**

Your datasets have data from 4 movements, each repeated 3 times by each of the six subjects as follows:

Four movement types (all with approximate 1Hz frequency, 10 repetitions/record, and within a vertical XZ plane parallel to, and in front of the subject’s torso):

1-CW circle

2-CW triangle

3-Linear, left to right

4-Linear, up-down

Signal sampling rate: 30Hz

Recoding device: on-wrist wearable with 3-axis accelerometer

**Overview:**

***For subject specific calculations:***Given three rounds of data, use a 3-fold cross validation and report on the ROC (copy the curve, quote its area or AUC, and equal error rate EER) as described below. To do so, first, produce all classifier output matrices as follows.

Fold 1: (subject specific) for ***all*** N movements (N=4), use run 1 and 2 for training the classifier, and then for each subject create NxN classifier output matrices (classifier continuous outputs without thresholding, a 4x4 matrix) using run 3 for each of the 6 subjects.

Fold 2: repeat using run 2 and 3 for training, 3 for testing.

Fold 3: repeat using run 1 and 3 for training, 2 for testing.

This should give you 3 classifier output matrices for each subject: H1, H2, and H3(one for each fold). Report on overall AUC and EER, and show 3-fold cross-validation ROC graph FOR EACH SUBJECT, using ezroc3(cat(3, H1, H2, H3)). Note: the diagonal of H matrices should be mostly positive, and off-diagonal elements should be mostly negative.

Repeat this for training, where you have two 4x4 H matrices per subject, one per each of the two runs used in training. Then report on average AUC and EER, and show ROC graph using ezroc3, e.g. using (cat(3, H1, H2, H3,H4, H5, H6)).

***For subject independent calculations:*** since we have 6 subjects, use a similar but 6-fold cross validation process where you train on 5 subjects and test on the sixth, and cycle trough 6 times and provide the 6-fold cross validation ROC curve accordingly (training not needed for subject independent).

**Specifics:**

Your inputs are the original x(t),z(t) signals of the 4 movements. First, produce all dynamic NN output matrices as follows (1 hidden layer, tansig-tansig MLP). You can use either msereg or trainbr, your choice. This part (especially for trainbr and NARX) is compute-intensive and time-consuming, so you need to start early. Read help files for TDNN and NARX very carefully, following their examples and the dynamic NN samples I ran for you in class. You can use decimate(x,n) to down-sample your signal x n times before using them should your computers freeze due to memory limitations.

**Task 1, TDNN:**

*Pre-fold 1:*First, using subject 2, use x(t) and z(t) from run 1 for neural net training, and run 2 (validation) for choosing configuration via ROC AUC. Use a TDNN with 4 outputs (+/-1 target signals with the same length as your x(t), y(t) for class/not-class). Try 3 different hidden layer sizes and 3 different input delay line depths. In case of weight decay, use regularized MSE by setting net.performFcn='msereg' with just one pressure value in the interest of time (set via net.performParam.ratio), for 3x3=9 configurations. As the starting point for the number hidden nodes, use 1/2 the size of input delay line. Compute the ensuing nine validation ROCs and their AUCs by stacking validation sim outputs of 5 initialized trainings (e.g. neti=init(net)) into a 4x4x5 H using ‘cat’ command, and don’t forget to save every trained network first! Train for 50 epochs for msereg and 100 for trainbr with no early stopping (i.e. set the train ratio to 1 and test and validation to 0). Choose the best hidden layer and tapped delay line sizes based on lowest validation ROC AUC before proceeding. In interest of time, use this best configuration for all the ensuing calculations. Report on your observations. Note: to convert the output of the dynamic neural nets from a time signal into a single value, replace each signal segment with its average. Don’t forget to use sequential rather than concurrent format when dealing with dynamic NNs. Next:

**Subject Specific TDNNs:**

*Fold 1:* use the above best TDNN configuration and create 6 individual output ‘H’ matrices using run 3 as test set for each of the 6 subjects (concatenate run 1 and 2 x(t) and z(t) into twice-long signals for training). Use no early stopping/validation. That is, using NN’s 4 outputs issued for 4 movements, once for each subject, to get 6 4x4 matrices. Again, you need to collapse each output time signal snippet relating to the period of execution of each movement into a scalar by averaging.

Repeat the above and find the three H matrices and stack them for three-fold validation by:

*Fold 2:*use run 2 and 3 for training, 1 for testing.

*Fold 3:* use run 1 and 3 for training, 2 for testing.

You will show 6 ROCs and their AUC & EERs as a result (one ROC per subject). No repeated initialization necessary (though you may optionally try the process say 3-5 times and show us the best result).

**Subject Specific NARX:**

This would be like the above, but you need to also find the output tap delay line length. As a rule of thumb, always use an output delay line shorter than the input. To reduce the number configurations to explore, find the best configuration by trying 2 input delay line depths x 2 hidden layer sizes x 3 output delay line depths. Initialize five times over fold 1 for subject two to find the best configuration, and the proceed as before.

Again, to find NN output, simply average all the samples over output period of interest. Setting initial conditions is necessary. Using NARX in series-parallel form (ONLY FOR TRAINING) or in native recurrent mode is your call, but close the loop for anything after training.

**For subject independent calculations (TDNN and NARX):** since we have 6 subjects, use a similar but 6-fold cross validation process where you train on 5 subjects and test on the sixth, and cycle trough 6 times and provide the training ROC curve accordingly.

Report on overall AUC and EER, and show ROC graph using ezroc3(cat (3, H1, H2, .... H6)). Note: the diagonal of H matrices should be mostly positive.

**Bonus:** create committees through sum rule (H-matrix averaging). Investigate committees for 6 subject-specific and the subject-independent cases (remember, the more diverse the committee, the better). Max 5 bonus points per each case upon successful completion (i.e. the resulting AUC better than that of the best constituent). Total maximum of 10 bonus points.

**Deliverables:** a power point, and all requested information above, including test ROC, AUC, EER, and your observations and conclusions + your code + trained NN objects.

Upload your code, objects, and PPT in a zip file with names as a part of filename (abbreviate if too long) on Blackboard.

Note: given the square decision profile matrices, the ezroc3 does not require target values T as a second input.

[ACC Data.zip](https://blackboard.umkc.edu/bbcswebdav/pid-2203387-dt-content-rid-9963146_1/xid-9963146_1)