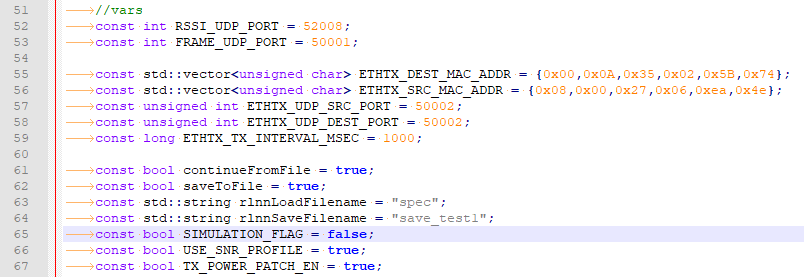
# Quick Guide to RLNN Software

**Prerequisites Libraries Required**: MLPack, Boost, Armadillo libraries

**Top Level CPP File**: rlnn4/RLNNCognitiveEngineTester\_v2.cpp

## 1.) Setting up System/Simulation Parameters

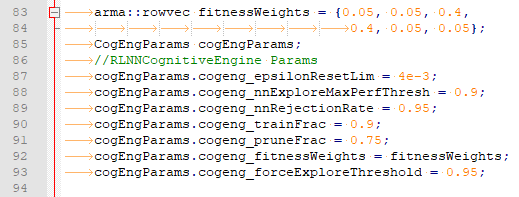


RSSI\_UDP\_PORT: ViaSat modem port for receiving RSSI updates  
FRAME\_UDP\_PORT: ViaSat modem port for receiving raw frames  
ETHTX\_UDP\_SRC\_PORT: Source port for sending to ML-605  
ETHTX\_UDP\_DEST\_PORT: Destination port for sending to ML-605  
ETHTX\_TX\_INTERVAL\_MSEC: deprecated

saveToFile: flag to save cognitive engine parameters to a file to be continued from in the future  
rlnnSaveFilename: filename of cognitive engine parameters to be saved  
continueFromFile: flag to continue cognitive engine from a previous run  
rlnnLoadFilename: filename of previously saved cognitive engine parameters

SIMULATION\_FLAG: False if using with ML-605/ViaSat or ARCC hardware. True if running without any hardware  
USE\_SNR\_PROFILE: True if using a custom SNR profile (named “snrProfile.txt”); False if using default linear ramp up and down profile. snrProfile.txt is a text file where each line is a EsN0 recording in dB.  
TX\_POWER\_PATCH\_EN: True to override the cognitive engine’s tuple decisions to only jump TX power by at most 1.5 dB between each tuple. False to disable.

## 2.) Setting up Cognitive Engine Parameters



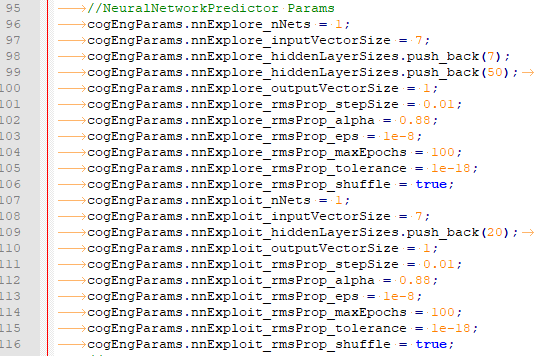
\_fitnessWeights: Multi-objective weight vector whose elements must sum to 1. The weights are {THROUGHPUT, BIT ERROR RATE, TARGET BANDWIDTH, SPECTRAL EFFICIENCY, TRANSMIT POWER EFFICIENCY, DC POWER CONSUMED}

\_epsilonResetLim: Explore/Exploit Epsilon minimum value. On each iteration, epsilon continues to decrease. It resets to when it reaches epsilonResetLim  
\_nnExploreMaxPerfThresh: when exploring, we threshold into two bins (good and bad actions). Good actions are defined as actions whose expected performance is \_nnExploreMaxPerfThresh\*max(performances of all possible actions).

\_nnRejectionRate: probability of choosing from the good bin of explorable actions.  
\_trainFrac: Fraction of the training set to use for training. 1-\_trainFrac is used for validation data  
\_pruneFrac: Fraction of the training buffer that we keep in the training buffer after a successful training process has occurred. (We eliminated 1-\_pruneFrac from the buffer)

\_forceExploreThreshold: if the performance difference between two consecutive recorded samples is >= \_forceExploreThreshold, then the cognitive engine is forced to throw away its buffer and re-explore the space.

## 2.) Setting up Neural Network Parameters (optional)



nnExplore\_nNets: number of parallel explore NNs in the Explore NN ensemble  
nnExplore\_inputVectorSize: number of neurons in the input layer of an Explore NN  
nnExplore\_hiddenLayerSizes: vector holding the number of neurons in each hidden layer of an Explore NN  
nnExplore\_outputVectorSize: number of neurons in the output layer of an Explore NN  
nnExplore\_rmsProp\_\*: deprecated

nnExploit\_nNets: number of parallel exploit NNs in the Exploit NN ensemble  
nnExploit\_inputVectorSize: number of neurons in the input layer of an Exploit NN  
nnExploit\_hiddenLayerSizes: vector holding the number of neurons in each hidden layer of an Exploit NN  
nnExploit\_outputVectorSize: number of neurons in the output layer of an Exploit NN  
nnExploit\_rmsProp\_\*: deprecated

## 3.) Setting up Application Specific Parameters



\_nOutVecFeatures: number of output vector features in training buffer (see Application Specific Module for details on these 8 features)  
\_frameSize: number of bits in DVB-S2 frame  
\_maxEsN0: maximum expected/possible EsN0 reading in run (used as a scaling factor) (in dB)  
\_minEsN0: minimum expected/possible EsN0 reading in run (used as a scaling factor) (in dB)  
\_modList: vector of number of symbols for each modCod pair  
\_codList: vector of code rate for each modCod pair  
\_modCodList: modCod ID numbers that correspond to the modCod pairs in \_modList and \_codList  
\_rollOffList: Allowable filter rolloff values  
\_symbolRateList: vector with all allowable symbol rates. On-the-fly symbol rate changes is not supported by DVB-S2. This vector should be a vector of one value for flight tests; for simulation, this can be any vector of symbol rates.  
\_transmitPowerList: vector of allowable relative transmit powers (dB).

## 4.) Setting up Training Buffer Parameters



\_nTrainTestSamples: Number of samples in the training buffer.

## 5.) Running the Code

Compile the code in Linux and then run it from a terminal (“./RLNNCognitiveEngineTester\_v2”). The output of the run is outputted to the terminal. A log file (logging.txt) is also written with more information.