

```
clc
clear all
```

Load the preprocessed dataset (stored in .mat files). Loaded data will be in the form of "struct" file. Extract the data inside that using dot indexing

```
Training = load('C:\courses\ML\project\codes\TrainNetworkProject2\dataset_processed\TwoclassTrain.mat');
Training = Training.Balancedsleep2ClassTrainUSCell;
TrainingLabels = load('C:\courses\ML\project\codes\TrainNetworkProject2\dataset_processed\TwoclassTrainLabels.mat');
TrainingLabels = TrainingLabels.TwoClassTrain_US_Labels';
Validation= load('C:\courses\ML\project\codes\TrainNetworkProject2\dataset_processed\TwoclassValidation.mat');
Validation = Validation.Balancedsleep2ClassValidationUSCell;
ValidationLabels = load('C:\courses\ML\project\codes\TrainNetworkProject2\dataset_processed\TwoclassValidationLabels.mat');
ValidationLabels = ValidationLabels.TwoClassValidation_US_Labels';
Testing= load('C:\courses\ML\project\codes\TrainNetworkProject2\dataset_processed\TwoclassTest.mat');
Testing = Testing.Balancedsleep2ClassTestUSCell;
TestingLabels = load('C:\courses\ML\project\codes\TrainNetworkProject2\dataset_processed\TwoclassTestLabels.mat');
TestingLabels = TestingLabels.TwoClassTest_US_Labels';
```

Hyperparameter tuning using Bayesian optimization approach (which is better than grid search and random search since it performs hyperparameter search in an "informed" fashion)

```
optimVars = [
    optimizableVariable('miniBatchSize',[2 128],'Type','integer')
    optimizableVariable('MaxEpochs',[10 60],'Type','integer')
    optimizableVariable('hiddenunits',[1000 3000],'Type','integer')
    optimizableVariable('InitialLearnRate',[1e-6 1e-2],'Transform','log')
    optimizableVariable('L2Regularization',[1e-10 1e-2],'Transform','log')
    optimizableVariable('dropout',[0.1 0.6])
    optimizableVariable('GradientThreshold',[0.1 3])
    optimizableVariable('GradientDecayFactor',[0.7 0.99])
    optimizableVariable('SquaredGradientDecayFactor',[0.7 0.99]));

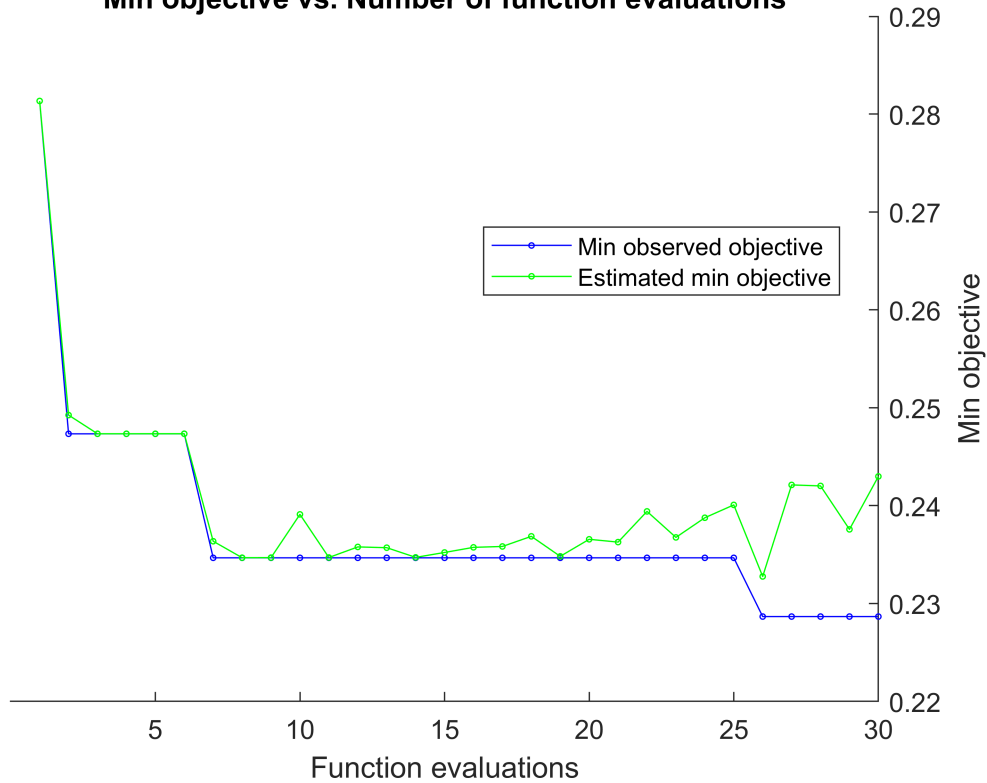
ObjFcn = makeObjFcn(Training,TrainingLabels,Validation,ValidationLabels);

BayesObject = bayesopt(ObjFcn,optimVars, ...
    'MaxTime',6*60*60, ...
    'IsObjectiveDeterministic',false, ...
    'UseParallel',false);
```

Iter	Eval result	Objective	Objective runtime	BestSoFar (observed)	BestSoFar (estim.)	miniBatchSize	MaxEpochs	hiddenunits
1	Best	0.28133	979.62	0.28133	0.28133	96	24	266
2	Best	0.24733	352.61	0.24733	0.24925	62	16	207
3	Accept	0.34667	128.99	0.24733	0.24734	82	38	105
4	Accept	0.33533	1105.9	0.24733	0.24734	13	43	166
5	Accept	0.29933	187.02	0.24733	0.24734	76	10	147

6	Accept	0.29	2657.5	0.24733	0.24734	49	44	1809
7	Best	0.23467	403.36	0.23467	0.23636	95	10	2587
8	Accept	0.26467	354.49	0.23467	0.23468	75	10	1887
9	Accept	0.242	1478.7	0.23467	0.23469	37	13	2999
10	Accept	0.254	443.5	0.23467	0.2391	111	10	2555
11	Accept	0.23867	398.72	0.23467	0.2347	123	11	2909
12	Accept	0.24733	517.64	0.23467	0.23578	106	15	2847
13	Accept	0.25467	606.46	0.23467	0.2357	76	44	2999
14	Accept	0.23667	485.96	0.23467	0.23471	91	14	2917
15	Accept	0.24533	1837.5	0.23467	0.23521	17	10	2737
16	Accept	0.25133	241.53	0.23467	0.23574	119	44	2577
17	Accept	0.242	121.49	0.23467	0.23583	117	11	1147
18	Accept	0.238	233.57	0.23467	0.23686	122	11	2987
19	Accept	0.24267	132.7	0.23467	0.23482	106	11	1537
20	Accept	0.30867	113.7	0.23467	0.23656	128	52	1697
=====								
Iter	Eval result	Objective	Objective runtime	BestSoFar (observed)	BestSoFar (estim.)	miniBatchSize	MaxEpochs	hiddenunits
=====								
21	Accept	0.252	373.93	0.23467	0.23627	69	11	2907
22	Accept	0.25467	84.226	0.23467	0.23941	116	11	1043
23	Accept	0.23933	209.96	0.23467	0.23675	125	15	2517
24	Accept	0.24667	220.11	0.23467	0.23876	102	11	2257
25	Accept	0.27133	146.32	0.23467	0.24007	121	59	1027
26	Best	0.22867	291.27	0.22867	0.23276	122	23	2787
27	Accept	0.26533	217.44	0.22867	0.24211	106	13	2387
28	Accept	0.27667	72.761	0.22867	0.24201	127	11	1147
29	Accept	0.43933	77.579	0.22867	0.23757	127	11	1277
30	Accept	0.25267	262.62	0.22867	0.24298	114	15	1727

Min objective vs. Number of function evaluations



Optimization completed.
MaxObjectiveEvaluations of 30 reached.
Total function evaluations: 30
Total elapsed time: 14756.8561 seconds

Total objective function evaluation time: 14737.1323

Best observed feasible point:

<u>miniBatchSize</u>	<u>MaxEpochs</u>	<u>hiddenunits</u>	<u>InitialLearnRate</u>	<u>L2Regularization</u>	<u>dropout</u>	<u>GradientThreshold</u>
122	23	2784	5.3572e-05	0.0012238	0.40049	1.7466

Observed objective function value = 0.22867

Estimated objective function value = 0.24324

Function evaluation time = 291.2681

Best estimated feasible point (according to models):

<u>miniBatchSize</u>	<u>MaxEpochs</u>	<u>hiddenunits</u>	<u>InitialLearnRate</u>	<u>L2Regularization</u>	<u>dropout</u>	<u>GradientThreshold</u>
122	11	2983	7.1915e-05	2.6044e-07	0.31799	2.7776

Estimated objective function value = 0.24298

Estimated function evaluation time = 234.3856

Support functions

```
function ObjFcn = makeObjFcn(XTrain,YTrain,XValidation,YValidation)
ObjFcn = @valErrorFun;
    function [valError,cons,fileName] = valErrorFun(optVars)

        numFeatures = size(XTrain{1},1);
        numClasses = numel(categories(YTrain));

        layers = [
            sequenceInputLayer(numFeatures,'Name','sequence')
            bilstmLayer(optVars.hiddenunits,'OutputMode','last','Name','bilstm')
            dropoutLayer(optVars.dropout,'Name','drop')
            fullyConnectedLayer(numClasses,'Name','fc')
            softmaxLayer('Name','softmax')
            classificationLayer('Name','classification')];
        % miniBatchSize = 16;

        numObservations = numel(XTrain);
        numIterationsPerEpoch = floor(numObservations / optVars.miniBatchSize);

        options = trainingOptions('adam', ...
            'GradientDecayFactor', optVars.GradientDecayFactor, ...
            'SquaredGradientDecayFactor', optVars.SquaredGradientDecayFactor, ...
            'MiniBatchSize',optVars.miniBatchSize, ...
            'InitialLearnRate',optVars.InitialLearnRate, ...
            'GradientThreshold',optVars.GradientThreshold, ...
            'Shuffle','every-epoch', ...
```

```

        'ValidationData',{XValidation,YValidation}, ...
        'ValidationFrequency',numIterationsPerEpoch, ...
            'L2Regularization',optVars.L2Regularization, ...
            'MaxEpochs', optVars.MaxEpochs, ...
            'ValidationPatience', 5,...
        'Plots','training-progress', ...
        'Verbose',false);

[netLSTM,info] = trainNetwork(XTrain,YTrain,layers,options);
close(findall(groot,'Tag','NNET_CNN_TRAININGPLOT_UIFIGURE'))

        YPredicted = classify(netLSTM,XValidation);
        valError = 1 - mean(YPredicted == YValidation);

        fileName = num2str(valError) + ".mat";
        save(fileName,'netLSTM','valError','options')
        cons = [];

    end
end

```