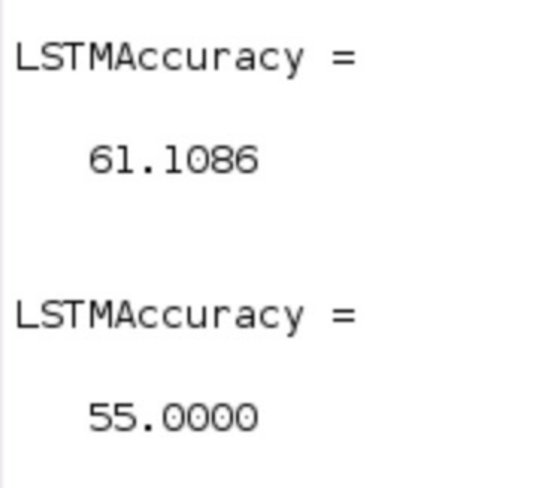
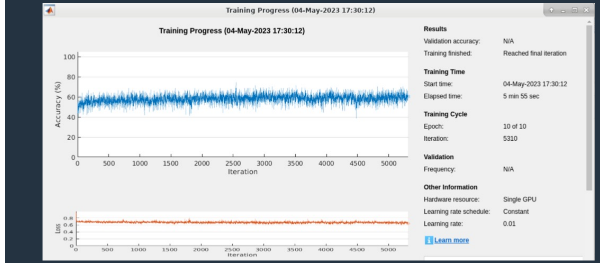
Mini Project 5

Emmitt Hasty

Default Settings



```
LSTMAccuracy =

   61.1086


LSTMAccuracy =

   55.0000
```



Training Progress (04-May-2023 17:30:12)

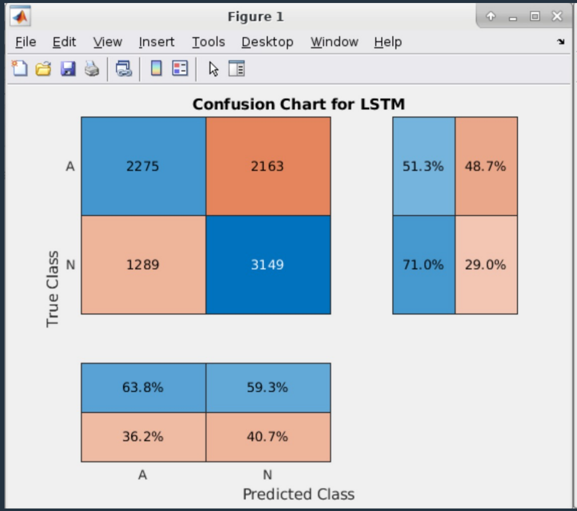
Results
Validation accuracy: N/A
Training finished: Reached final iteration

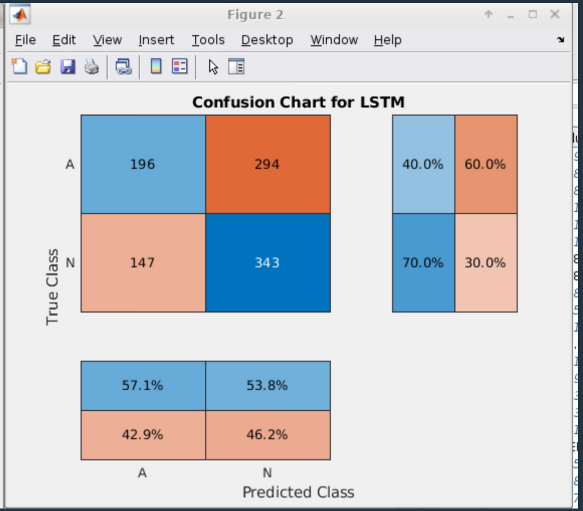
Training Time
Start time: 04-May-2023 17:30:12
Elapsed time: 5 min 55 sec

Training Cycle
Epoch: 10 of 10
Iteration: 5310

Validation
Frequency: N/A

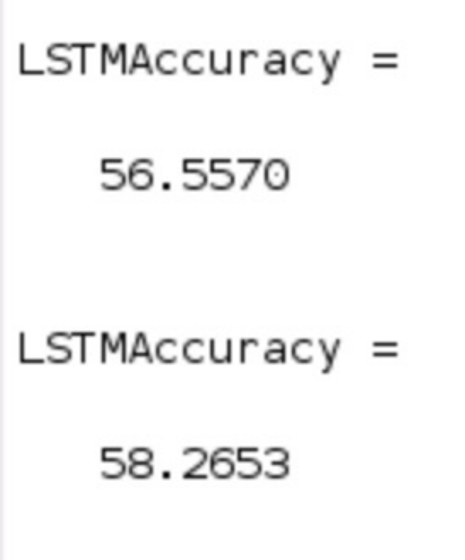
Other Information
Hardware resource: Single GPU
Learning rate schedule: Constant
Learning rate: 0.01



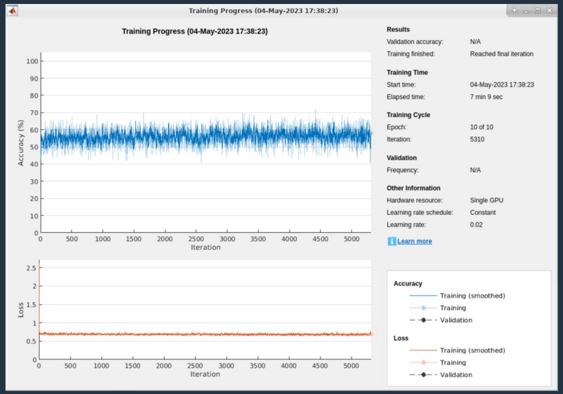


Variant 1

→ Changed it to 150 layers, and a learning rate of 0.02. It did not yield better results. It was slightly more accurate on the test data than the default model.



```
LSTMAccuracy =

   56.5570


LSTMAccuracy =

   58.2653
```



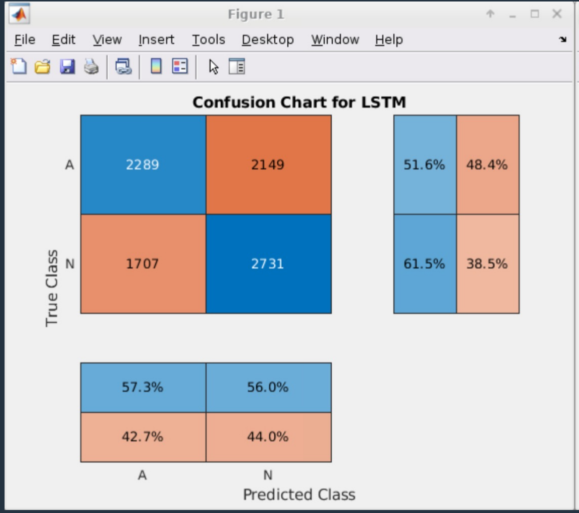
Results
Validation accuracy: N/A
Training finished: Reached final iteration

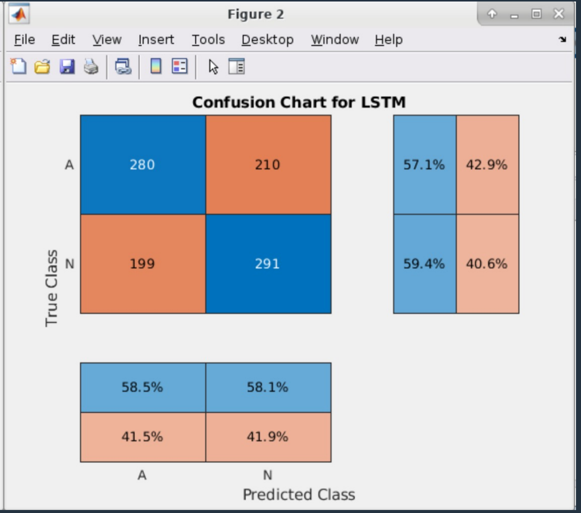
Training Time
Start time: 04-May-2023 17:38:23
Elapsed time: 7 min 9 sec

Training Cycle
Epoch: 10 of 10
Iteration: 5310

Validation
Frequency: N/A

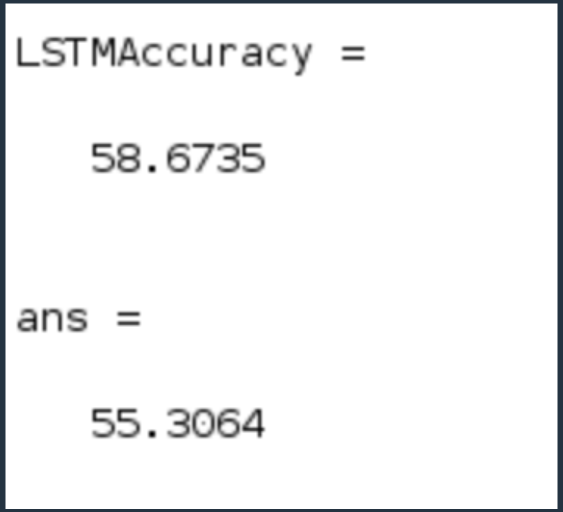
Other Information
Hardware resource: Single GPU
Learning rate schedule: Constant
Learning rate: 0.02



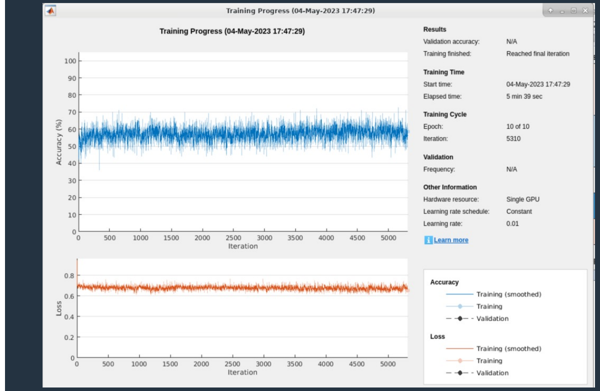


Variant 2

→ This one was brought down to 75 layers, and returned to a learning rate of 0.01. It did not work at all. It led to a .3 increase in the test accuracy, which is not significant.

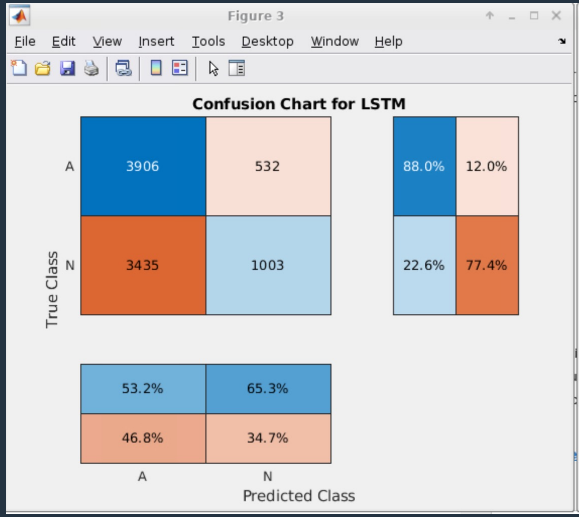


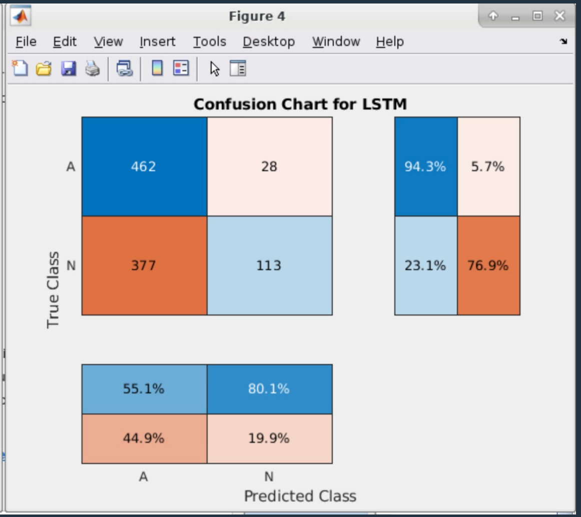
```
LSTMAccuracy =

   58.6735


ans =

   55.3064
```



Training Progress (04-May-2023 17:47:29)

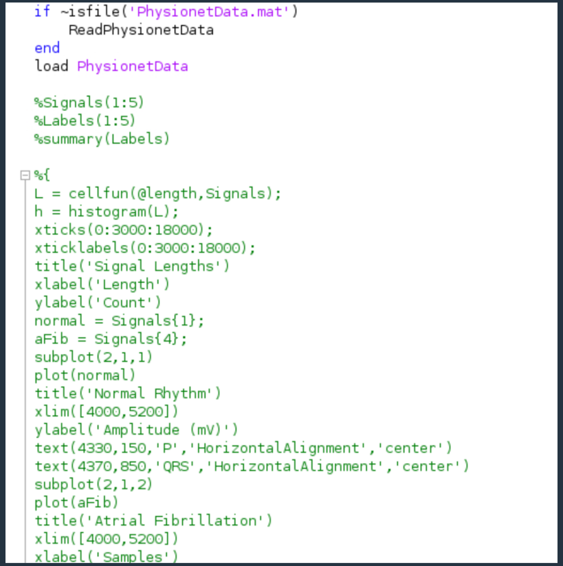
```
Results
Validation accuracy:        N/A
Training finished:          Reached final iteration

Training Time
Start time:                 04-May-2023 17:47:29
Elapsed time:               5 min 39 sec

Training Cycle
Epoch:                      10 of 10
Iteration:                  5310

Validation
Frequency:                  N/A

Other Information
Hardware resource:          Single GPU
Learning rate schedule:     Constant
Learning rate:              0.01
```



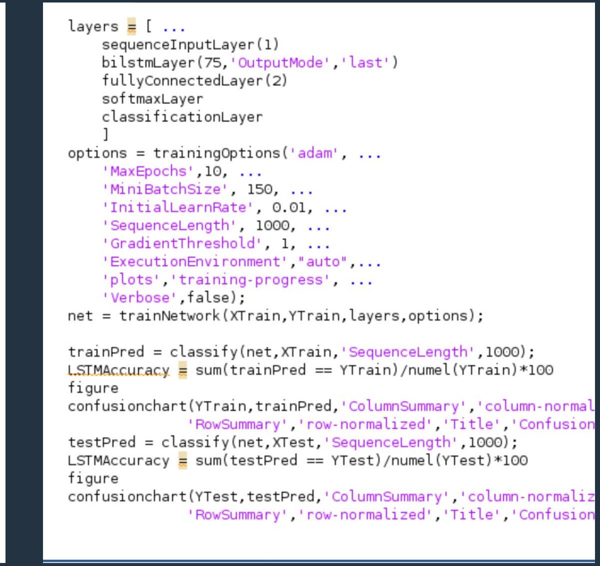


Code

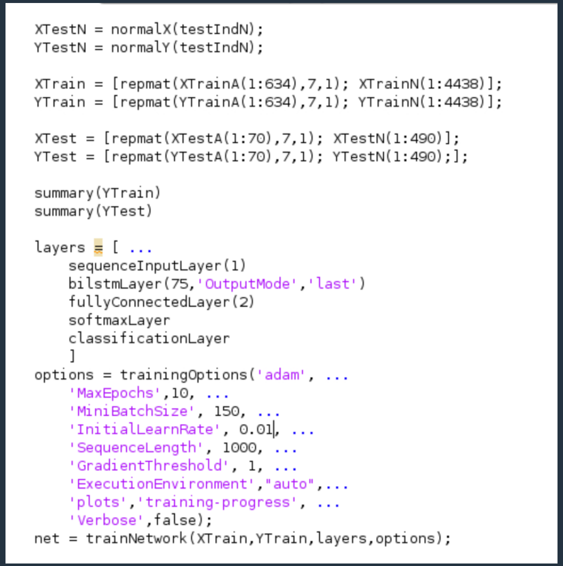
→ Commented out some of the original code from the documentation.



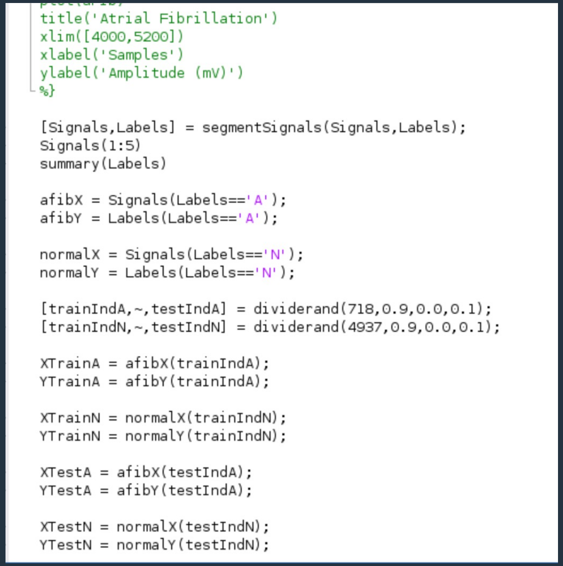
```
if ~isfile('PhysionetData.mat')
    ReadPhysionetData
end
load PhysionetData

%Signals(1:5)
%Labels(1:5)
%summary(Labels)

%{
L = cellfun(@length,Signals);
h = histogram(L);
xticks(0:3000:18000);
xticklabels(0:3000:18000);
title('Signal Lengths')
xlabel('Length')
ylabel('Count')
normal = Signals{1};
aFib = Signals{4};
subplot(2,1,1)
plot(normal)
title('Normal Rhythm')
xlim([4000,5200])
ylabel('Amplitude (mV)')
text(4330,150,'P','HorizontalAlignment','center')
text(4370,850,'QRS','HorizontalAlignment','center')
subplot(2,1,2)
plot(aFib)
title('Atrial Fibrillation')
xlim([4000,5200])
xlabel('Samples')
```



```
layers = [ ...
    sequenceInputLayer(1)
    bilstmLayer(75,'OutputMode','last')
    fullyConnectedLayer(2)
    softmaxLayer
    classificationLayer
    ]
options = trainingOptions('adam', ...
    'MaxEpochs',10, ...
    'MiniBatchSize', 150, ...
    'InitialLearnRate', 0.01, ...
    'SequenceLength', 1000, ...
    'GradientThreshold', 1, ...
    'ExecutionEnvironment',"auto",...
    'plots','training-progress', ...
    'Verbose',false);
net = trainNetwork(XTrain,YTrain,layers,options);

trainPred = classify(net,XTrain,'SequenceLength',1000);
LSTMAccuracy = sum(trainPred == YTrain)/numel(YTrain)*100
figure
confusionchart(YTrain,trainPred,'ColumnSummary','column-normal
              'RowSummary','row-normalized','Title','Confusion
testPred = classify(net,XTest,'SequenceLength',1000);
LSTMAccuracy = sum(testPred == YTest)/numel(YTest)*100
figure
confusionchart(YTest,testPred,'ColumnSummary','column-normaliz
              'RowSummary','row-normalized','Title','Confusion
```



```
XTestN = normalX(testIndN);
YTestN = normalY(testIndN);

XTrain = [repmat(XTrainA(1:634),7,1); XTrainN(1:4438)];
YTrain = [repmat(YTrainA(1:634),7,1); YTrainN(1:4438)];

XTest = [repmat(XTestA(1:70),7,1); XTestN(1:490)];
YTest = [repmat(YTestA(1:70),7,1); YTestN(1:490);];

summary(YTrain)
summary(YTest)

layers = [ ...
    sequenceInputLayer(1)
    bilstmLayer(75,'OutputMode','last')
    fullyConnectedLayer(2)
    softmaxLayer
    classificationLayer
    ]
options = trainingOptions('adam', ...
    'MaxEpochs',10, ...
    'MiniBatchSize', 150, ...
    'InitialLearnRate', 0.01, ...
    'SequenceLength', 1000, ...
    'GradientThreshold', 1, ...
    'ExecutionEnvironment',"auto",...
    'plots','training-progress', ...
    'Verbose',false);
net = trainNetwork(XTrain,YTrain,layers,options);
```



```
plot(aFib)
title('Atrial Fibrillation')
xlim([4000,5200])
xlabel('Samples')
ylabel('Amplitude (mV)')
%}

[Signals,Labels] = segmentSignals(Signals,Labels);
Signals(1:5)
summary(Labels)

afibX = Signals(Labels=='A');
afibY = Labels(Labels=='A');

normalX = Signals(Labels=='N');
normalY = Labels(Labels=='N');

[trainIndA,~,testIndA] = dividerand(718,0.9,0.0,0.1);
[trainIndN,~,testIndN] = dividerand(4937,0.9,0.0,0.1);

XTrainA = afibX(trainIndA);
YTrainA = afibY(trainIndA);

XTrainN = normalX(trainIndN);
YTrainN = normalY(trainIndN);

XTestA = afibX(testIndA);
YTestA = afibY(testIndA);

XTestN = normalX(testIndN);
YTestN = normalY(testIndN);
```