Cough Segmentation

remove previous data

add yamnet to serach path

net =

SeriesNetwork with properties:

Layers: [86×1 nnet.cnn.layer.Layer]

InputNames: {'input\_1'}

OutputNames: {'Sound'}

86×1 Layer array with layers:

1 'input\_1' Image Input 96×64×1 images

2 'conv2d' Convolution 32 3×3×1 convolutions with stride [2 2] and padding 'same'

3 'b' Batch Normalization Batch normalization with 32 channels

4 'activation' ReLU ReLU

5 'depthwise\_conv2d' Grouped Convolution 32 groups of 1 3×3×1 convolutions with stride [1 1] and padding 'same'

6 'L11' Batch Normalization Batch normalization with 32 channels

7 'activation\_1' ReLU ReLU

8 'conv2d\_1' Convolution 64 1×1×32 convolutions with stride [1 1] and padding 'same'

9 'L12' Batch Normalization Batch normalization with 64 channels

10 'activation\_2' ReLU ReLU

11 'depthwise\_conv2d\_1' Grouped Convolution 64 groups of 1 3×3×1 convolutions with stride [2 2] and padding 'same'

12 'L21' Batch Normalization Batch normalization with 64 channels

13 'activation\_3' ReLU ReLU

14 'conv2d\_2' Convolution 128 1×1×64 convolutions with stride [1 1] and padding 'same'

15 'L22' Batch Normalization Batch normalization with 128 channels

16 'activation\_4' ReLU ReLU

17 'depthwise\_conv2d\_2' Grouped Convolution 128 groups of 1 3×3×1 convolutions with stride [1 1] and padding 'same'

18 'L31' Batch Normalization Batch normalization with 128 channels

19 'activation\_5' ReLU ReLU

20 'conv2d\_3' Convolution 128 1×1×128 convolutions with stride [1 1] and padding 'same'

21 'L32' Batch Normalization Batch normalization with 128 channels

22 'activation\_6' ReLU ReLU

23 'depthwise\_conv2d\_3' Grouped Convolution 128 groups of 1 3×3×1 convolutions with stride [2 2] and padding 'same'

24 'L41' Batch Normalization Batch normalization with 128 channels

25 'activation\_7' ReLU ReLU

26 'conv2d\_4' Convolution 256 1×1×128 convolutions with stride [1 1] and padding 'same'

27 'L42' Batch Normalization Batch normalization with 256 channels

28 'activation\_8' ReLU ReLU

29 'depthwise\_conv2d\_4' Grouped Convolution 256 groups of 1 3×3×1 convolutions with stride [1 1] and padding 'same'

30 'L51' Batch Normalization Batch normalization with 256 channels

31 'activation\_9' ReLU ReLU

32 'conv2d\_5' Convolution 256 1×1×256 convolutions with stride [1 1] and padding 'same'

33 'L52' Batch Normalization Batch normalization with 256 channels

34 'activation\_10' ReLU ReLU

35 'depthwise\_conv2d\_5' Grouped Convolution 256 groups of 1 3×3×1 convolutions with stride [2 2] and padding 'same'

36 'L61' Batch Normalization Batch normalization with 256 channels

37 'activation\_11' ReLU ReLU

38 'conv2d\_6' Convolution 512 1×1×256 convolutions with stride [1 1] and padding 'same'

39 'L62' Batch Normalization Batch normalization with 512 channels

40 'activation\_12' ReLU ReLU

41 'depthwise\_conv2d\_6' Grouped Convolution 512 groups of 1 3×3×1 convolutions with stride [1 1] and padding 'same'

42 'L71' Batch Normalization Batch normalization with 512 channels

43 'activation\_13' ReLU ReLU

44 'conv2d\_7' Convolution 512 1×1×512 convolutions with stride [1 1] and padding 'same'

45 'L72' Batch Normalization Batch normalization with 512 channels

46 'activation\_14' ReLU ReLU

47 'depthwise\_conv2d\_7' Grouped Convolution 512 groups of 1 3×3×1 convolutions with stride [1 1] and padding 'same'

48 'L81' Batch Normalization Batch normalization with 512 channels

49 'activation\_15' ReLU ReLU

50 'conv2d\_8' Convolution 512 1×1×512 convolutions with stride [1 1] and padding 'same'

51 'L82' Batch Normalization Batch normalization with 512 channels

52 'activation\_16' ReLU ReLU

53 'depthwise\_conv2d\_8' Grouped Convolution 512 groups of 1 3×3×1 convolutions with stride [1 1] and padding 'same'

54 'L91' Batch Normalization Batch normalization with 512 channels

55 'activation\_17' ReLU ReLU

56 'conv2d\_9' Convolution 512 1×1×512 convolutions with stride [1 1] and padding 'same'

57 'L92' Batch Normalization Batch normalization with 512 channels

58 'activation\_18' ReLU ReLU

59 'depthwise\_conv2d\_9' Grouped Convolution 512 groups of 1 3×3×1 convolutions with stride [1 1] and padding 'same'

60 'L101' Batch Normalization Batch normalization with 512 channels

61 'activation\_19' ReLU ReLU

62 'conv2d\_10' Convolution 512 1×1×512 convolutions with stride [1 1] and padding 'same'

63 'L102' Batch Normalization Batch normalization with 512 channels

64 'activation\_20' ReLU ReLU

65 'depthwise\_conv2d\_10' Grouped Convolution 512 groups of 1 3×3×1 convolutions with stride [1 1] and padding 'same'

66 'L111' Batch Normalization Batch normalization with 512 channels

67 'activation\_21' ReLU ReLU

68 'conv2d\_11' Convolution 512 1×1×512 convolutions with stride [1 1] and padding 'same'

69 'L112' Batch Normalization Batch normalization with 512 channels

70 'activation\_22' ReLU ReLU

71 'depthwise\_conv2d\_11' Grouped Convolution 512 groups of 1 3×3×1 convolutions with stride [2 2] and padding 'same'

72 'L121' Batch Normalization Batch normalization with 512 channels

73 'activation\_23' ReLU ReLU

74 'conv2d\_12' Convolution 1024 1×1×512 convolutions with stride [1 1] and padding 'same'

75 'L122' Batch Normalization Batch normalization with 1024 channels

76 'activation\_24' ReLU ReLU

77 'depthwise\_conv2d\_12' Grouped Convolution 1024 groups of 1 3×3×1 convolutions with stride [1 1] and padding 'same'

78 'L131' Batch Normalization Batch normalization with 1024 channels

79 'activation\_25' ReLU ReLU

80 'conv2d\_13' Convolution 1024 1×1×1024 convolutions with stride [1 1] and padding 'same'

81 'L132' Batch Normalization Batch normalization with 1024 channels

82 'activation\_26' ReLU ReLU

83 'global\_average\_pooling2d' Global Average Pooling Global average pooling

84 'dense' Fully Connected 521 fully connected layer

85 'softmax' Softmax softmax

86 'Sound' Classification Output crossentropyex with 'Speech' and 520 other classes

Create audioDatastore object of the data and split it into train and validation sets

Create audioFeatureExtractor object from audio signals

**win\_length** **overlap** **num\_spectrums** **segment\_length**

**\_\_\_\_\_\_\_\_\_\_** **\_\_\_\_\_\_\_** **\_\_\_\_\_\_\_\_\_\_\_\_\_** **\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**samples** 128 0.5 96 6208

**sec**  0.008 NaN NaN 0.388

Extract features from train and validation sets

100%

100%

cough segmentation requires only 2 classes (cough / non-cough).

Read in YAMNet, convert it to a [layerGraph](docid:nnet_ref#mw_399d64ed-680b-46f0-af77-0472ba1910d8), and then replace the final [fullyConnectedLayer](docid:nnet_ref#mw_1e7fbc56-4746-4f30-8cd9-7048ce806a0d) and the final [classificationLayer](docid:nnet_ref#bu5lho8) to reflect the new task.

Define training options

Train and save network

Training on single GPU.

|======================================================================================================================|

| Epoch | Iteration | Time Elapsed | Mini-batch | Validation | Mini-batch | Validation | Base Learning |

| | | (hh:mm:ss) | Accuracy | Accuracy | Loss | Loss | Rate |

|======================================================================================================================|

| 1 | 1 | 00:00:04 | 63.28% | 76.92% | 0.6296 | 0.5432 | 0.0003 |

| 1 | 37 | 00:00:32 | 97.27% | 92.02% | 0.0878 | 0.2425 | 0.0003 |

| 2 | 50 | 00:00:42 | 97.27% | | 0.0717 | | 3.0000e-05 |

| 2 | 74 | 00:01:01 | 98.44% | 92.15% | 0.0463 | 0.2393 | 3.0000e-05 |

| 3 | 100 | 00:01:21 | 100.00% | | 0.0167 | | 3.0000e-06 |

| 3 | 111 | 00:01:32 | 98.44% | 91.84% | 0.0413 | 0.2438 | 3.0000e-06 |

| 4 | 148 | 00:02:01 | 97.66% | 91.71% | 0.0608 | 0.2519 | 3.0000e-07 |

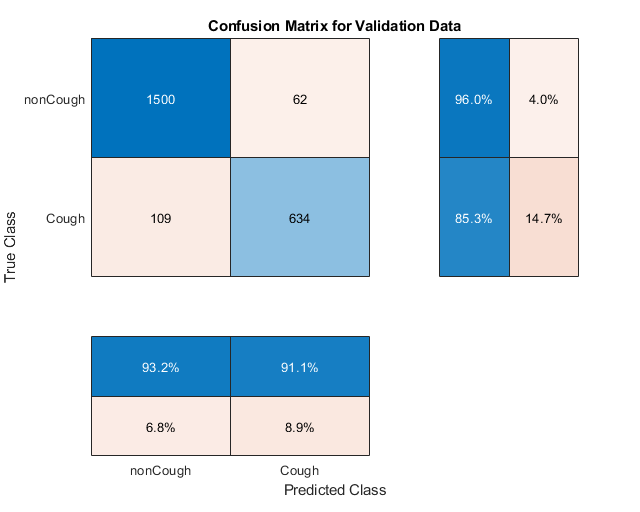
| 5 | 150 | 00:02:03 | 98.83% | | 0.0362 | | 3.0000e-08 |

| 5 | 185 | 00:02:31 | 98.05% | 92.19% | 0.0472 | 0.2494 | 3.0000e-08 |

|======================================================================================================================|



Use [confusionchart](https://www.mathworks.com/help/deeplearning/ref/confusionchart.html) (Deep Learning Toolbox) to evaluate the performance of the network on the validation set.

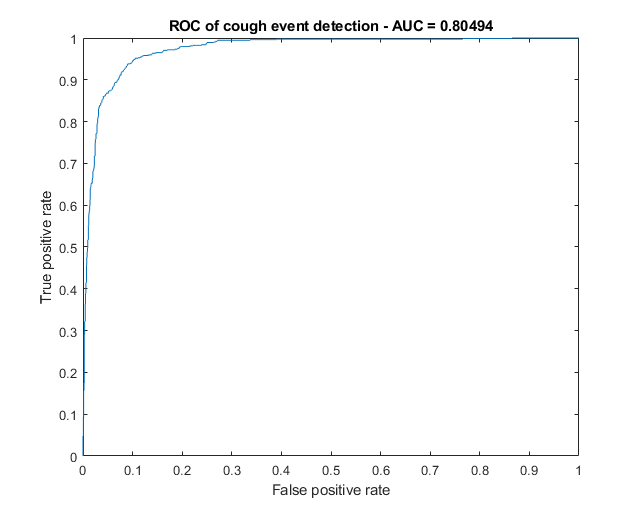


**accuracy [%]** **sensitivity [%]** **PPV [%]** **F1\_score [%]**

**\_\_\_\_\_\_\_\_\_\_\_\_** **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** **\_\_\_\_\_\_\_** **\_\_\_\_\_\_\_\_\_\_\_\_**

92.6 85.3 91.1 88.1

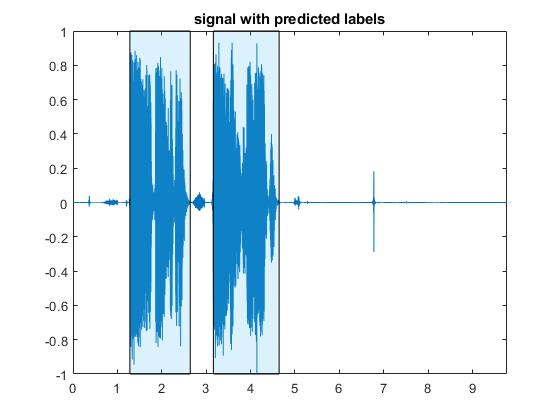
ROC



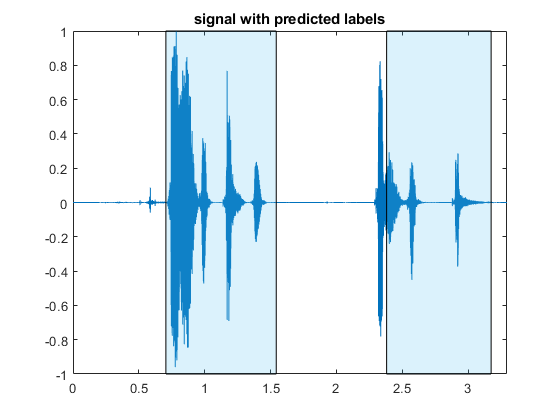
Save the trained network

plot segmentation results on validation data set

100%



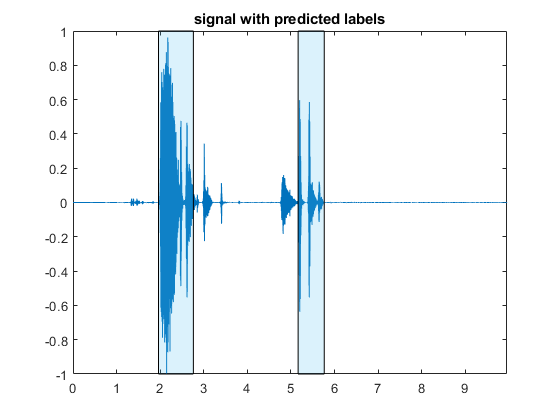
1



2



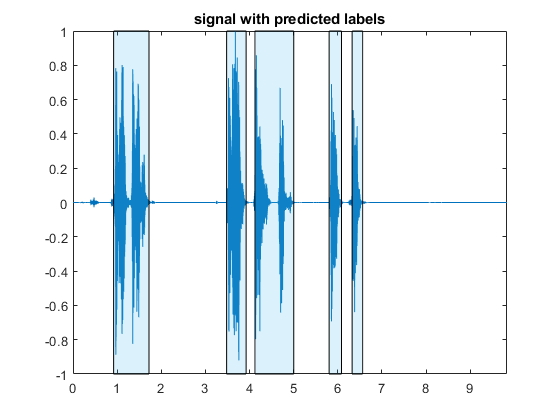
3



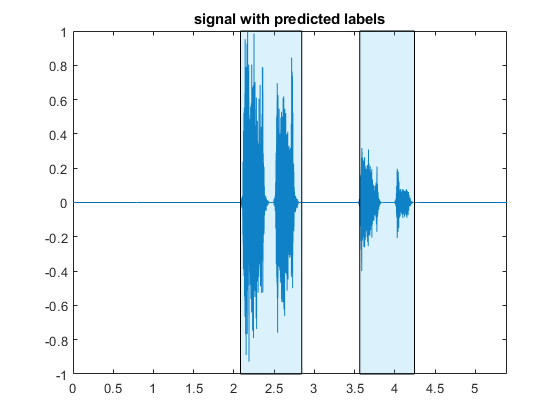
4



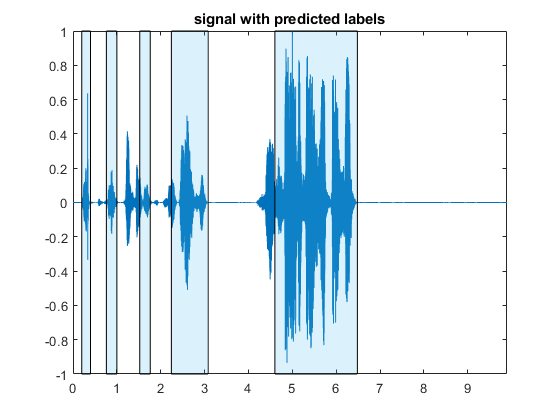
5



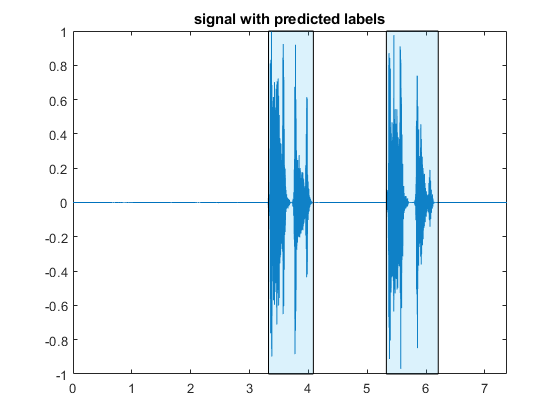
6



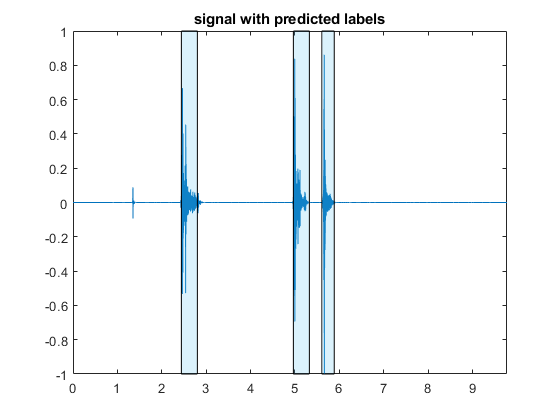
7



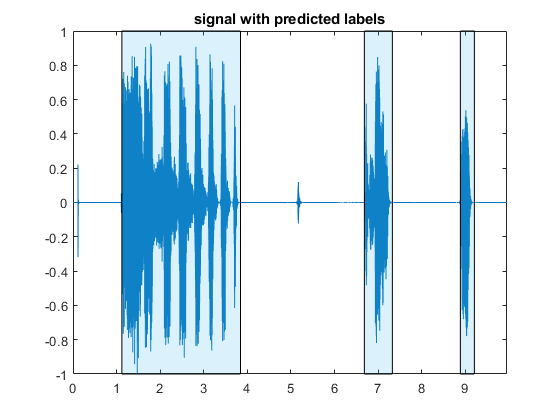
8



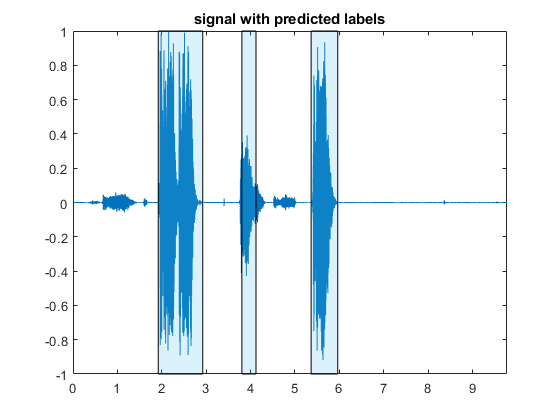
9



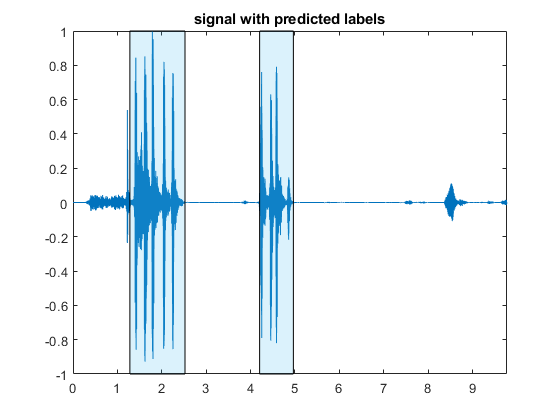
10



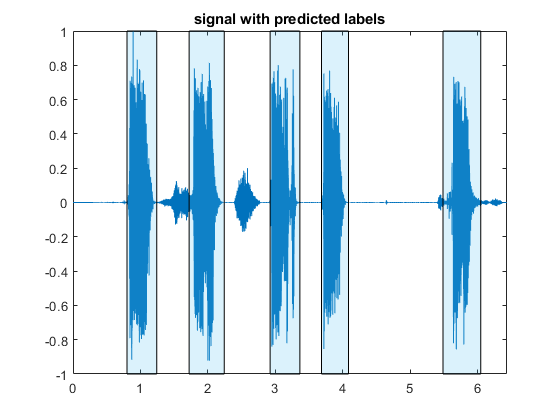
11



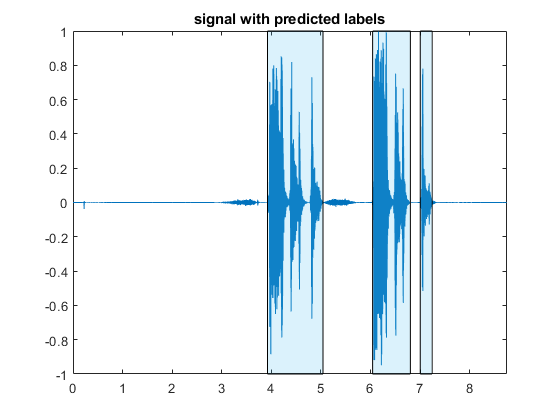
12



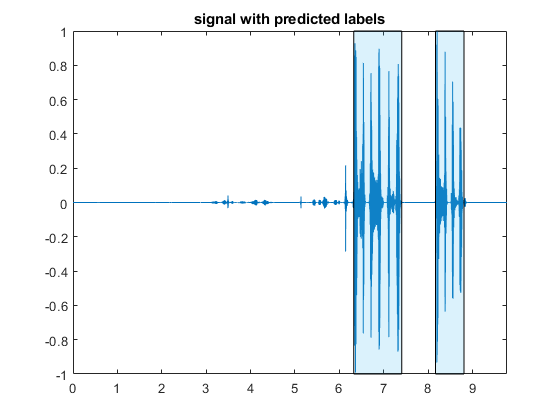
13



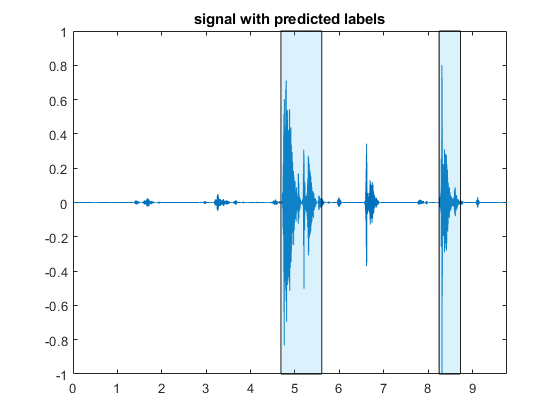
14



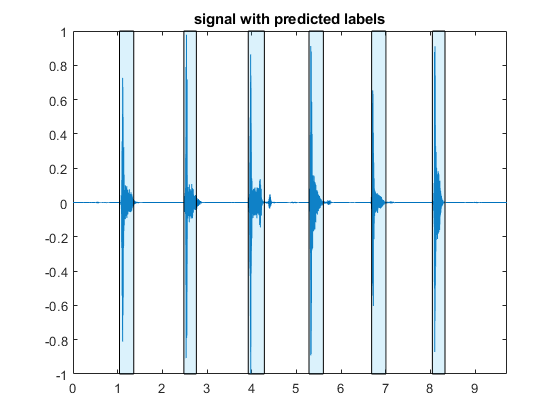
15



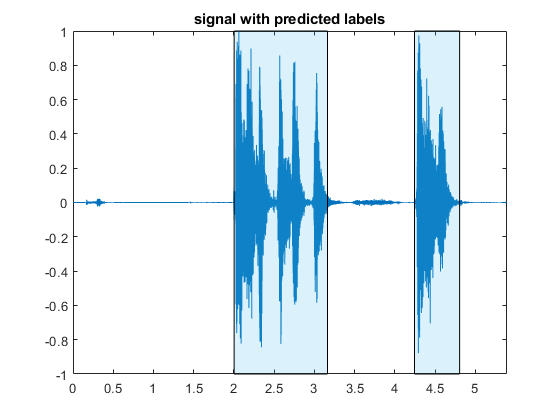
16



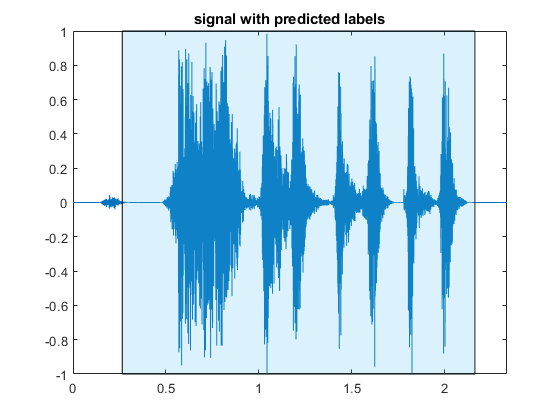
17



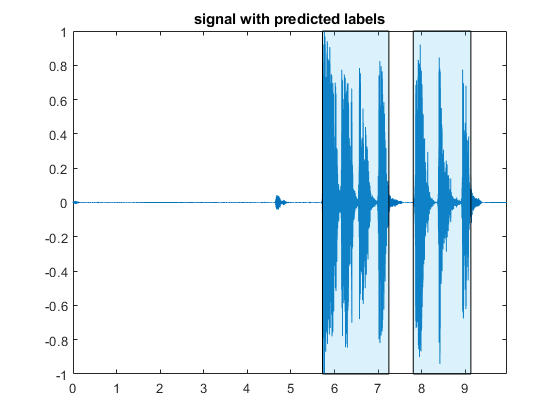
18



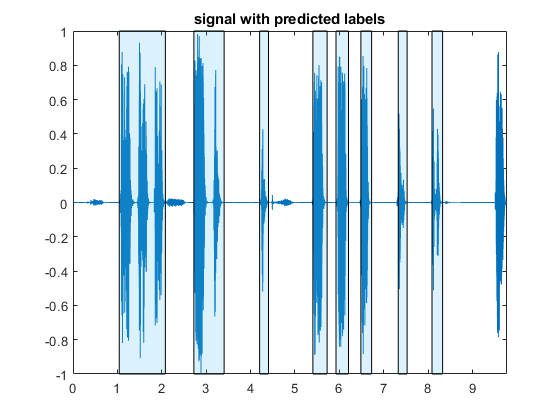
19



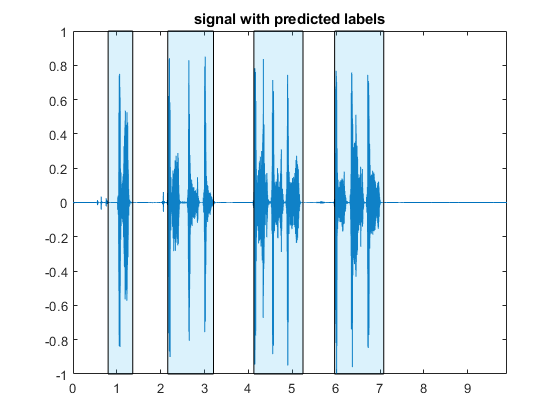
20



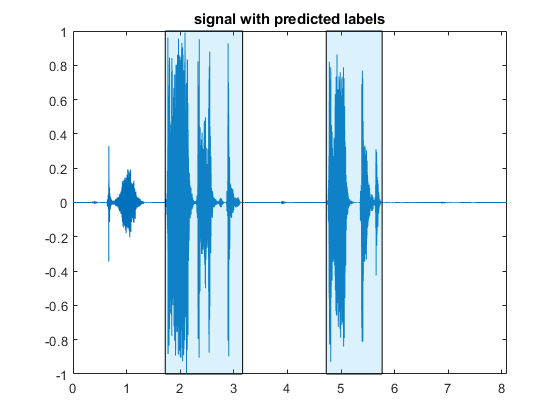
21



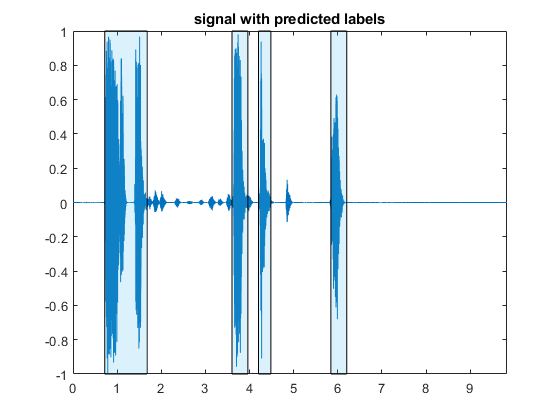
22



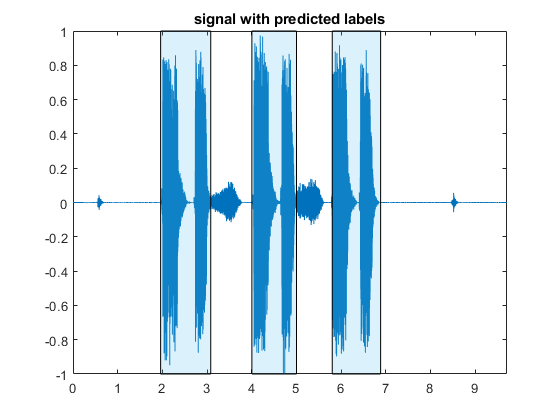
23



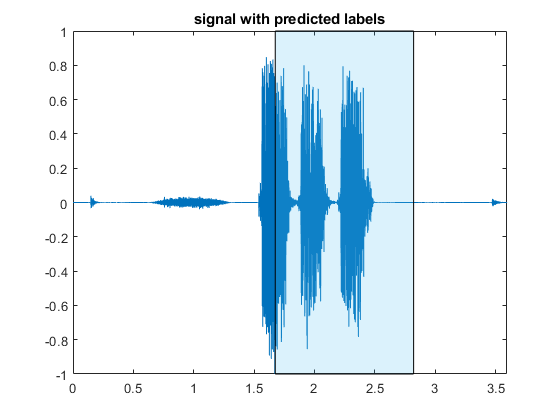
24



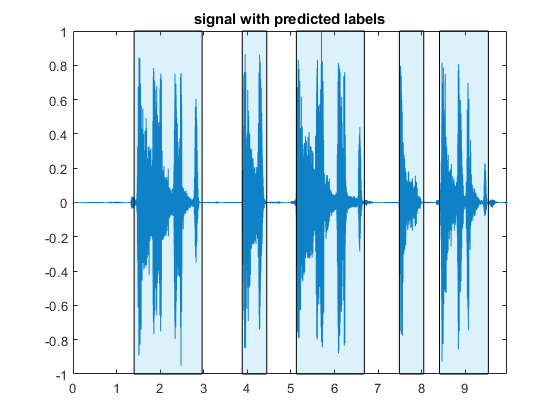
25



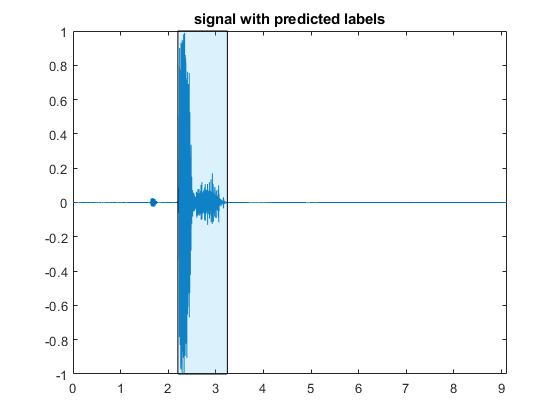
26



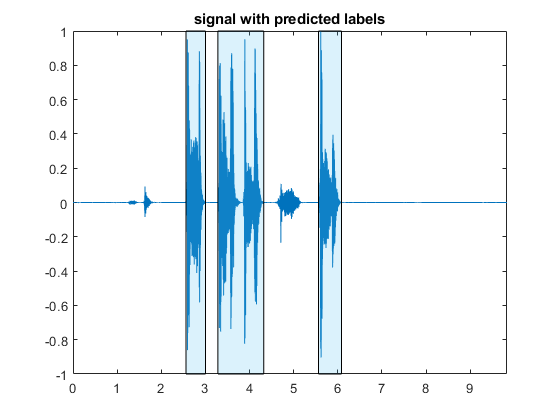
27



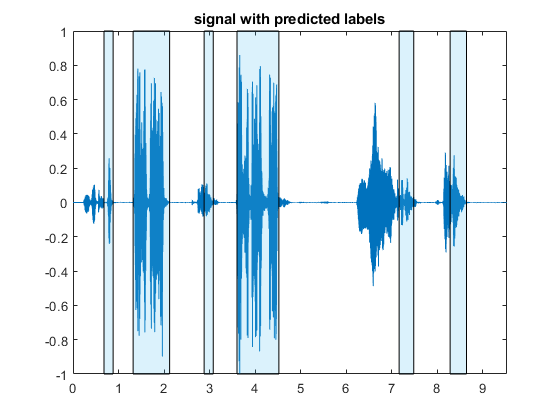
28



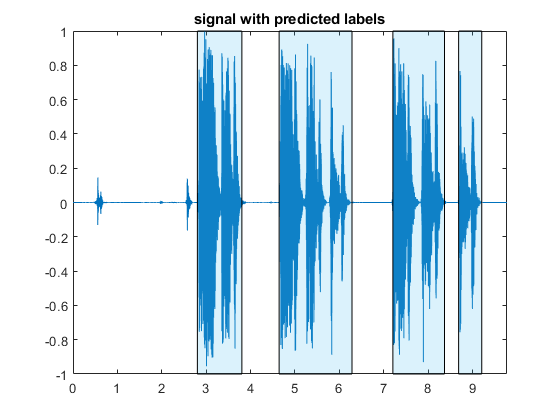
29



30



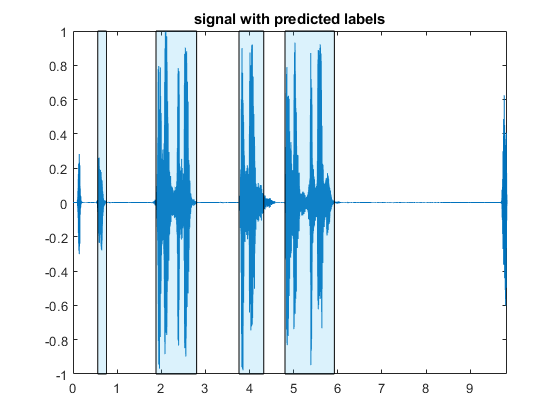
31



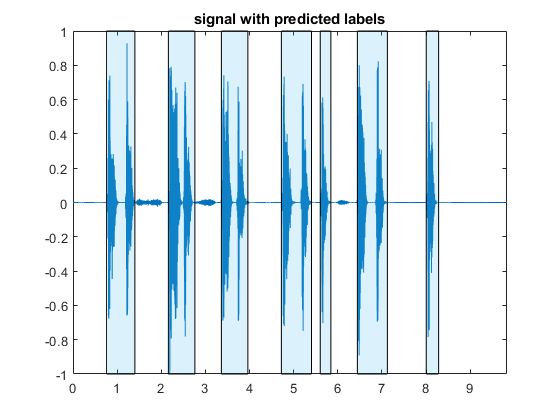
32



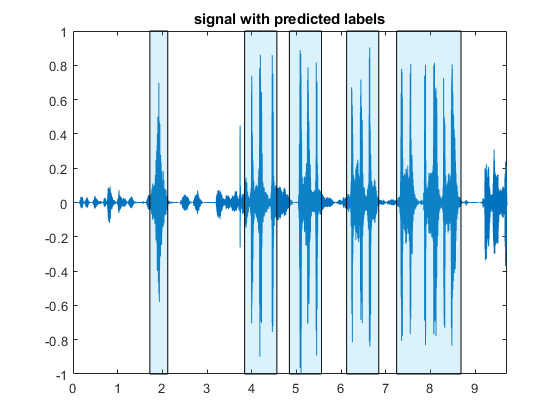
33



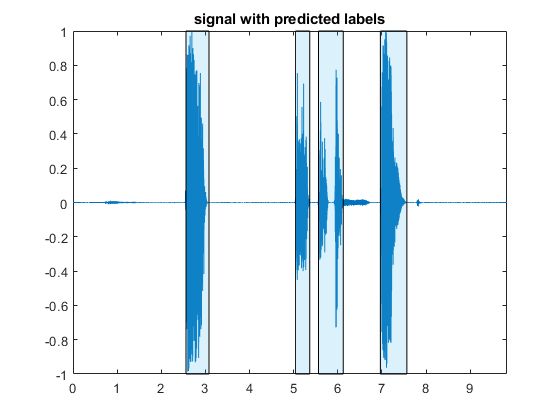
34



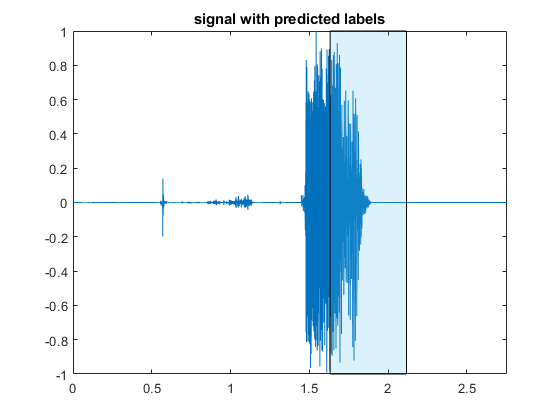
35



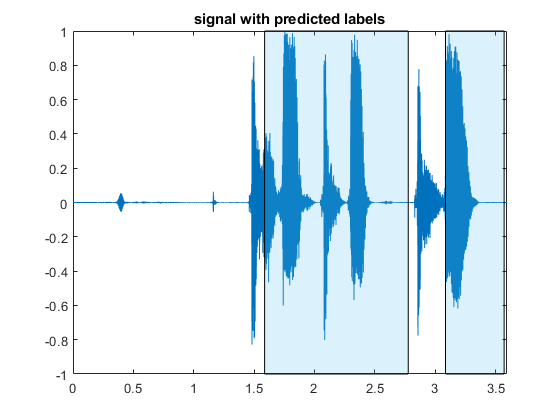
36



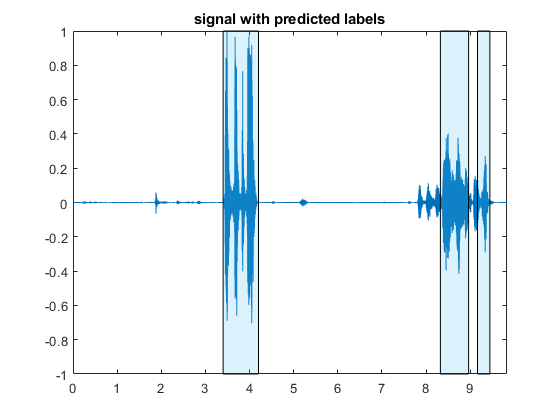
37



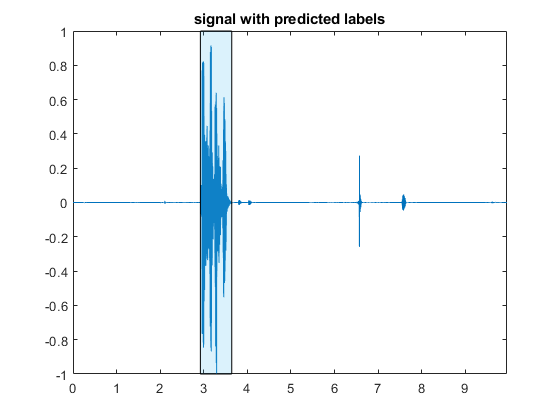
38



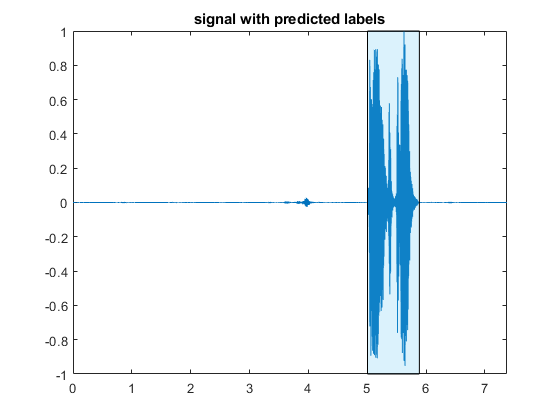
39



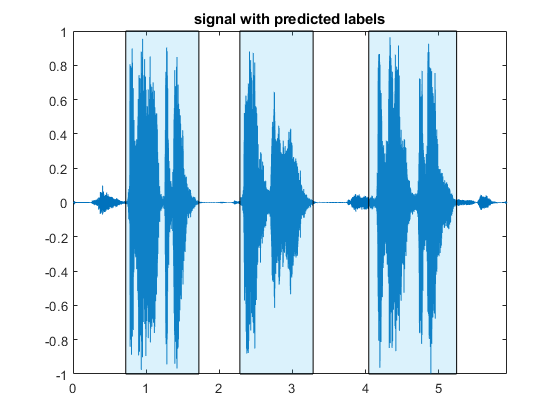
40



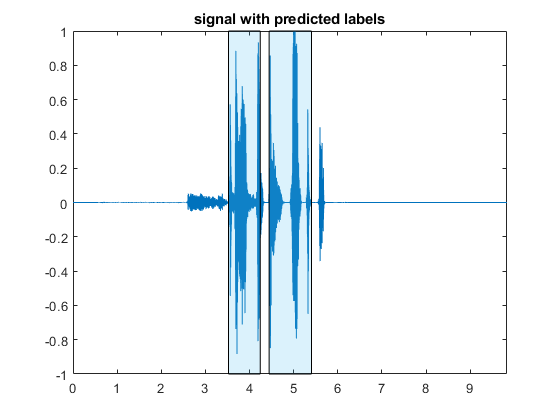
41



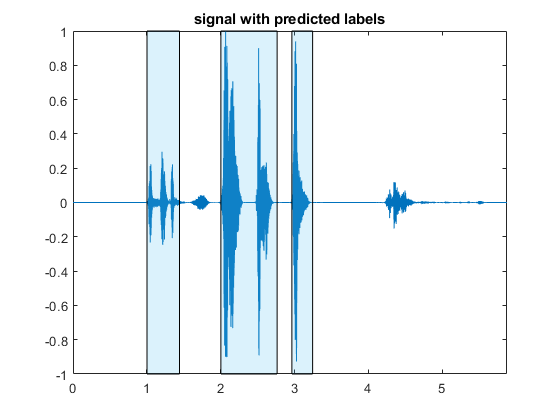
42



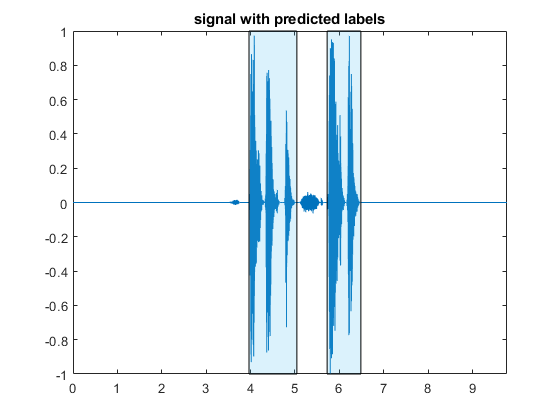
43



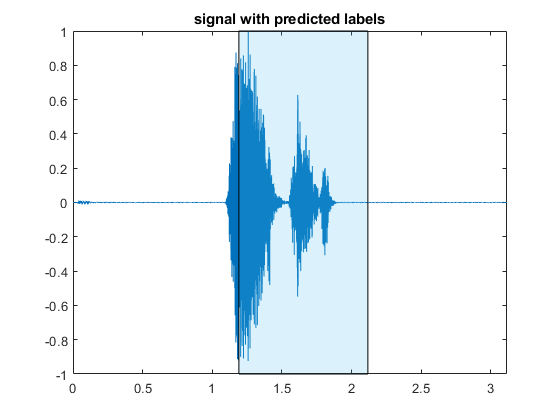
44



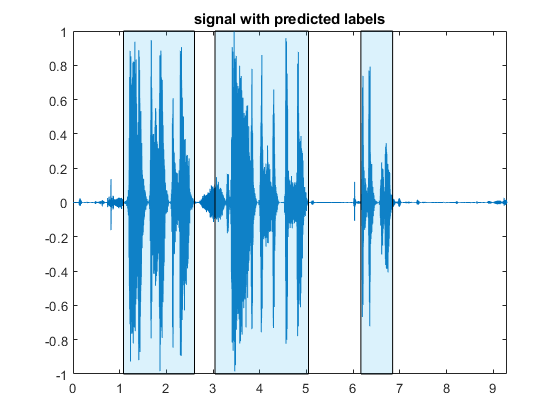
45



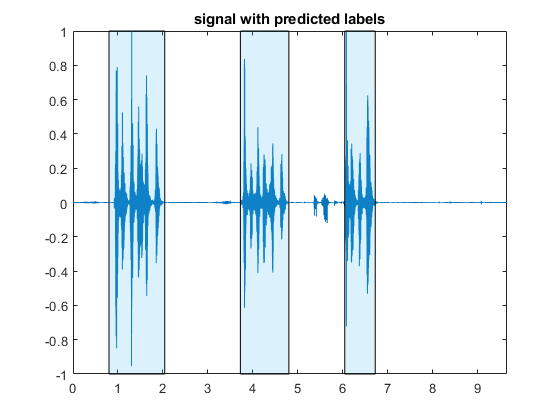
46



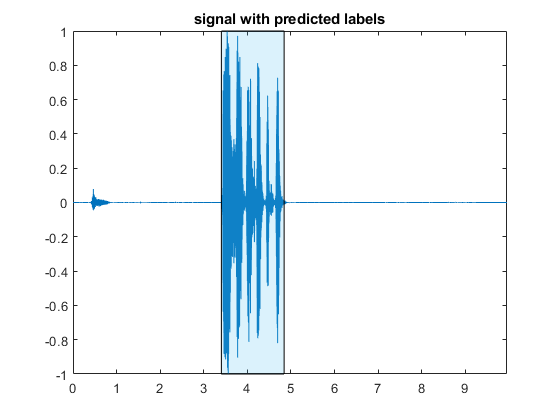
47



48



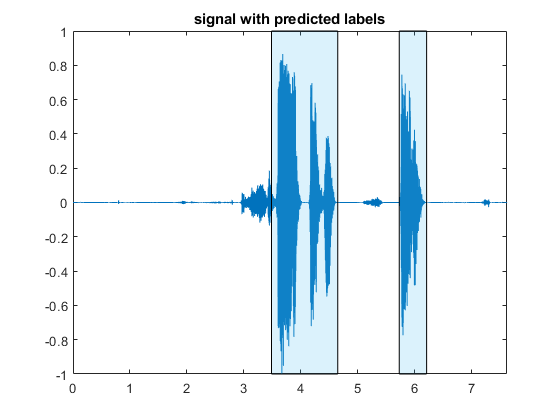
49



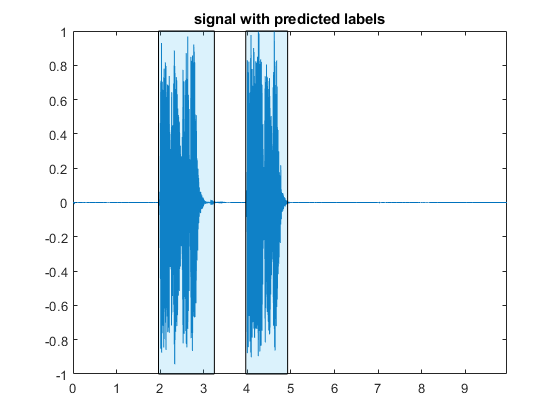
50



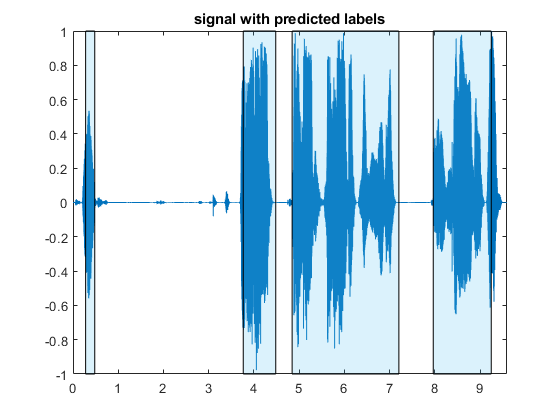
51



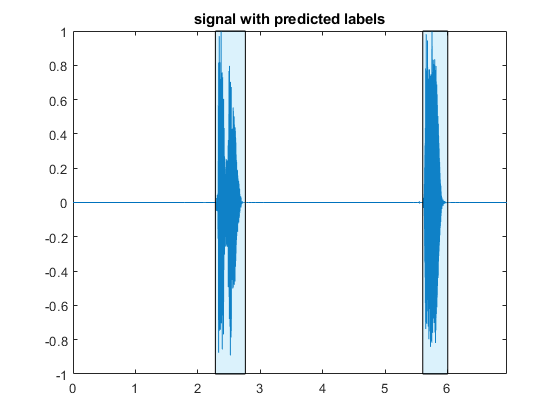
52



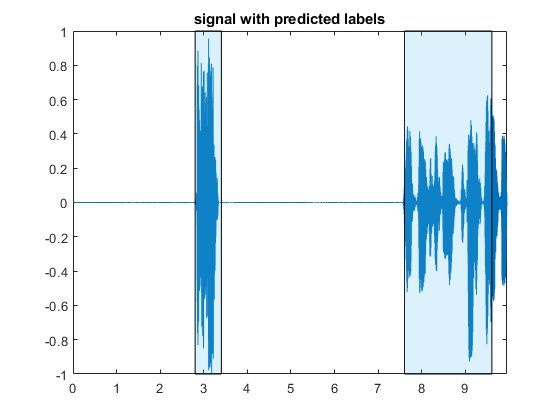
53



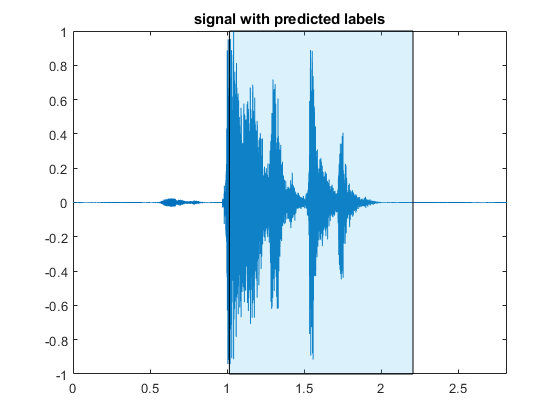
54



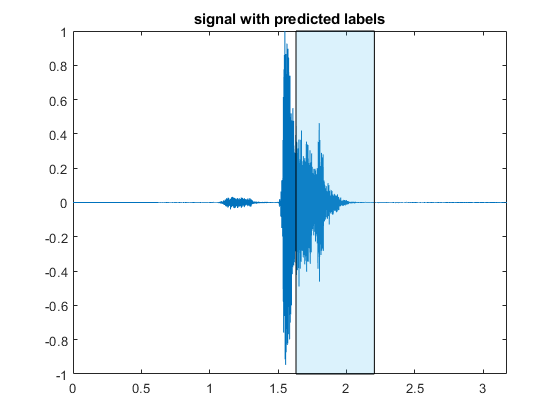
55



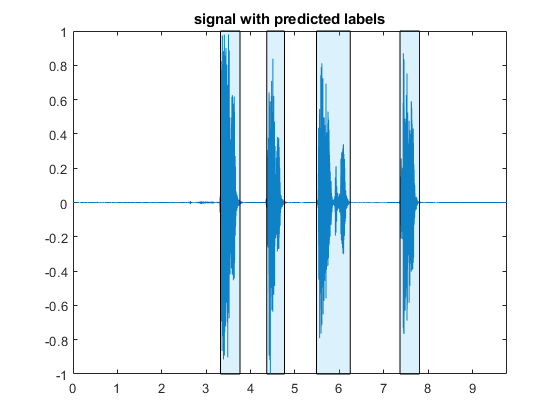
56



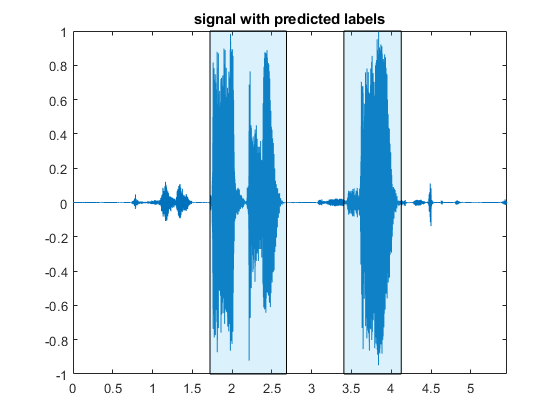
57



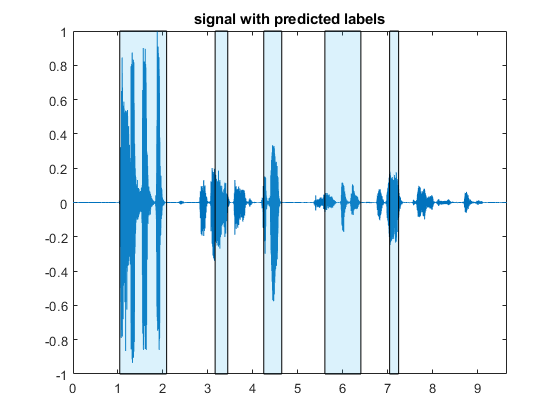
58



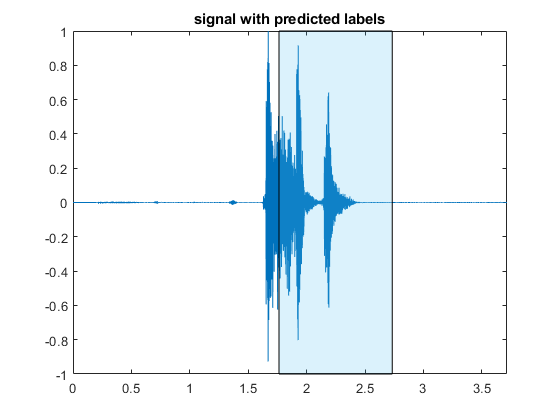
59



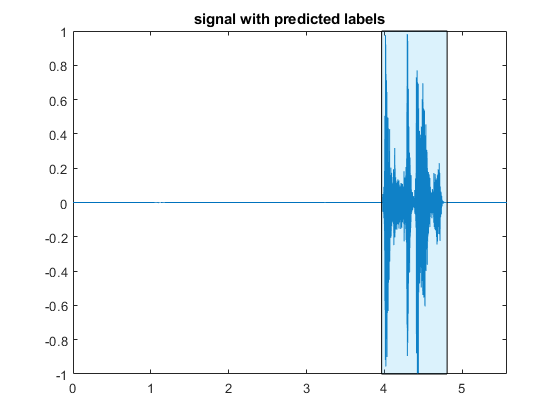
60



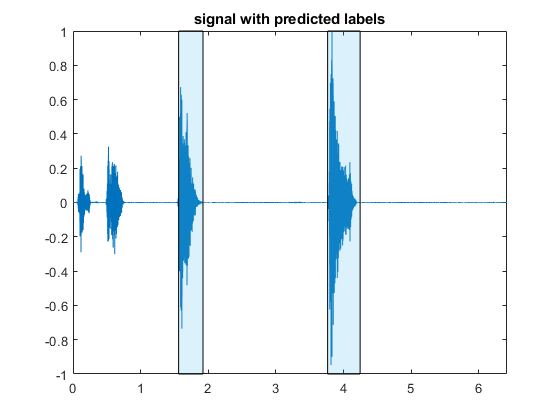
61



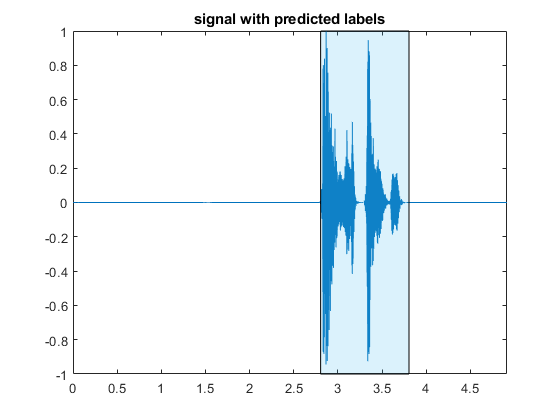
62



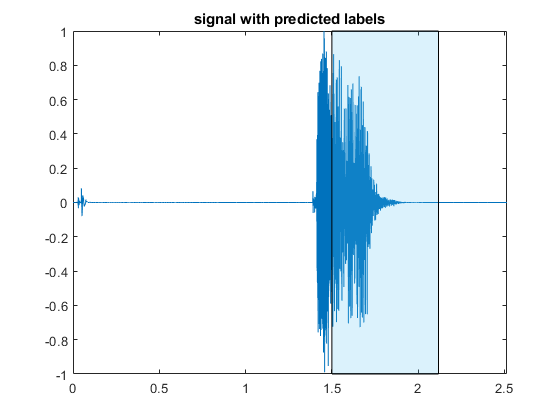
63



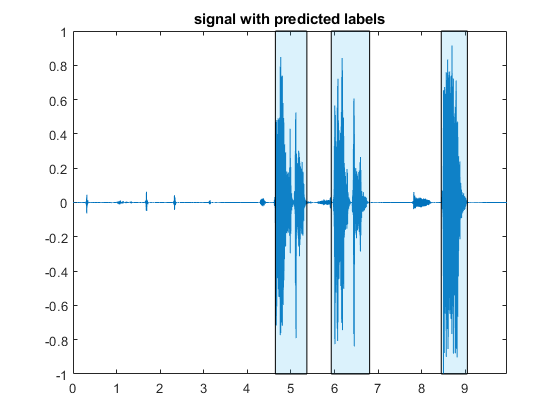
64



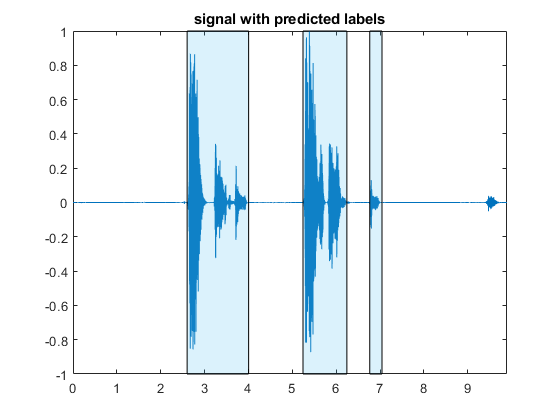
65



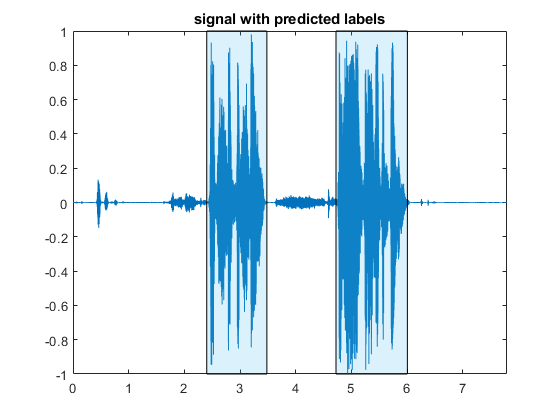
66



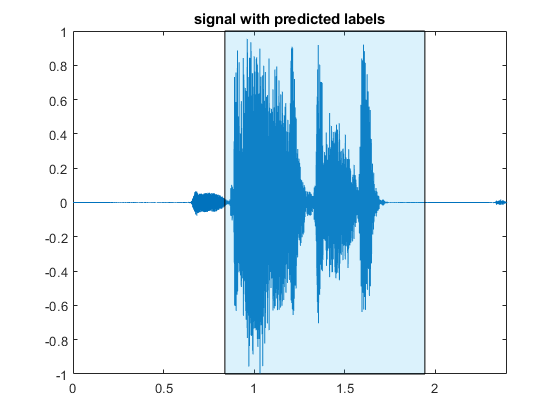
67



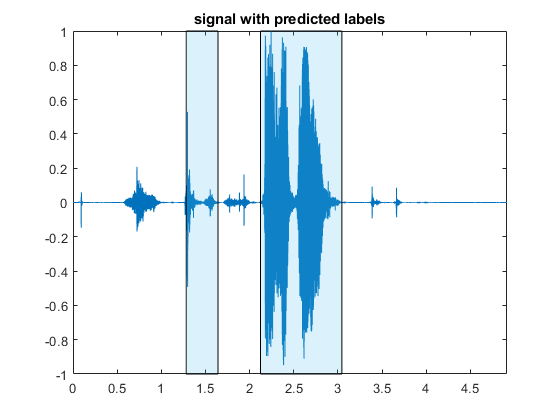
68



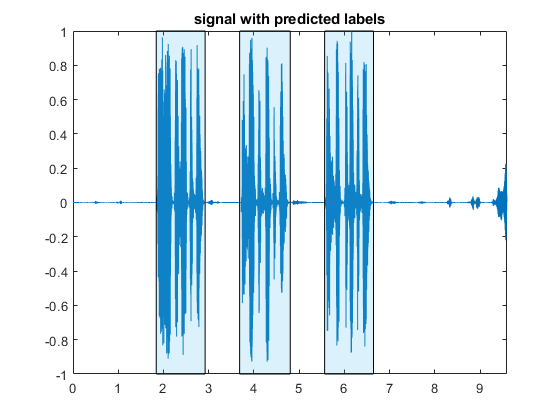
69



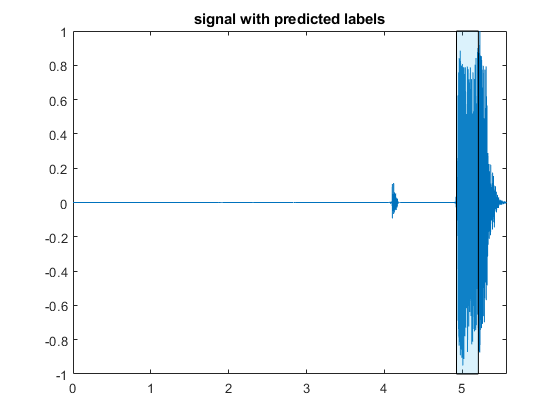
70



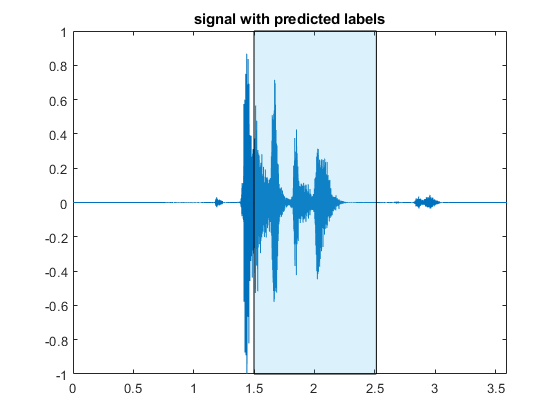
71



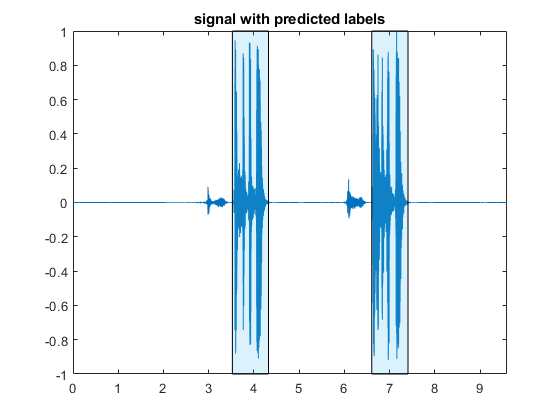
72



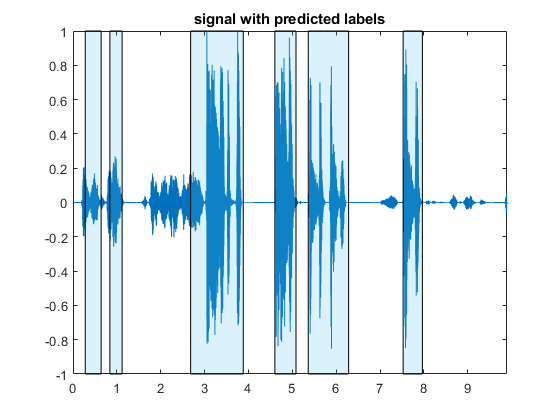
73



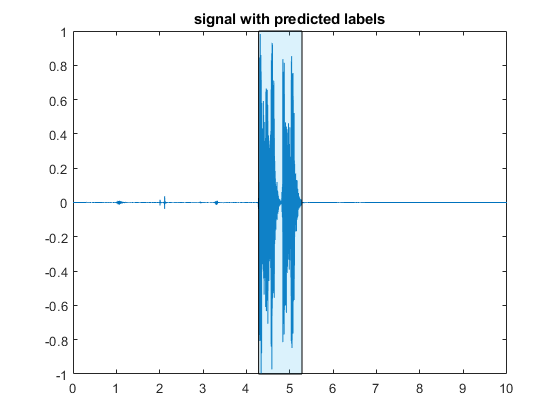
74



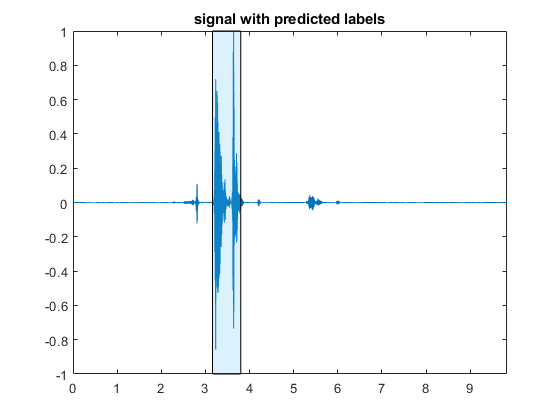
75



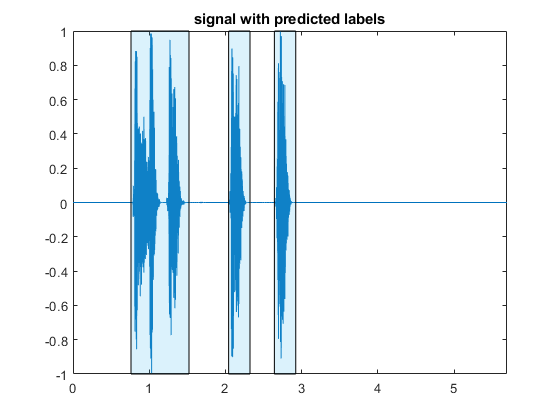
76



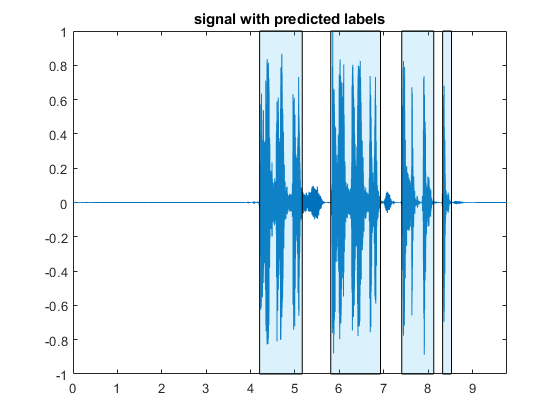
77



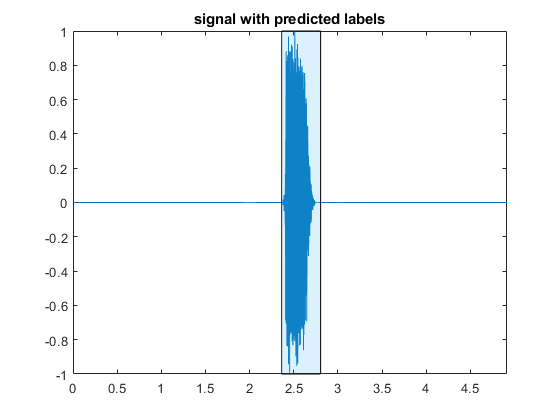
78



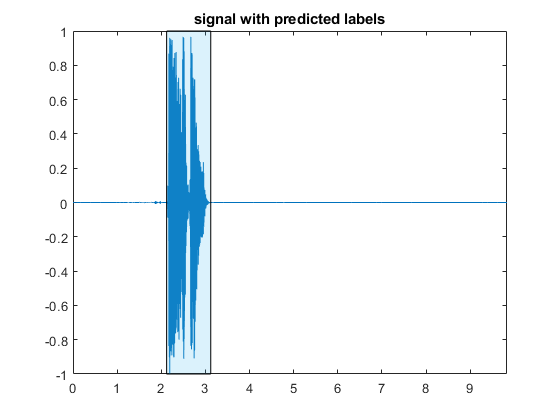
79



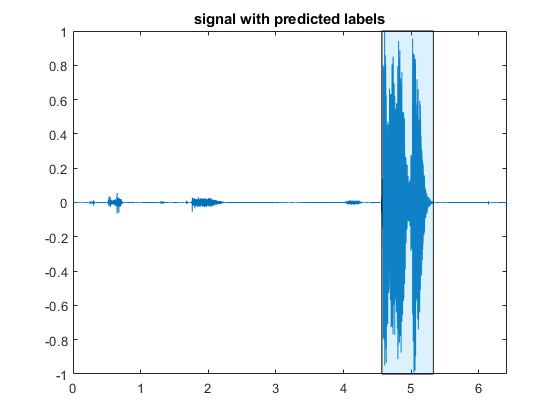
80



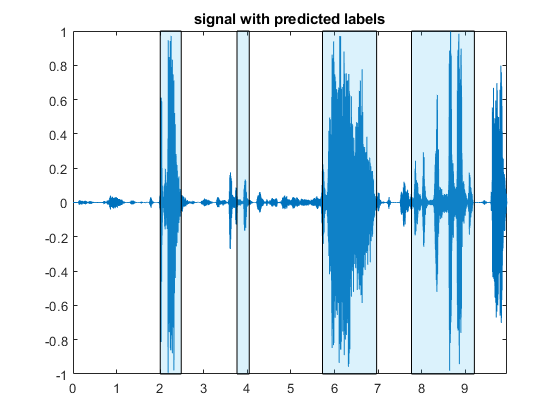
81



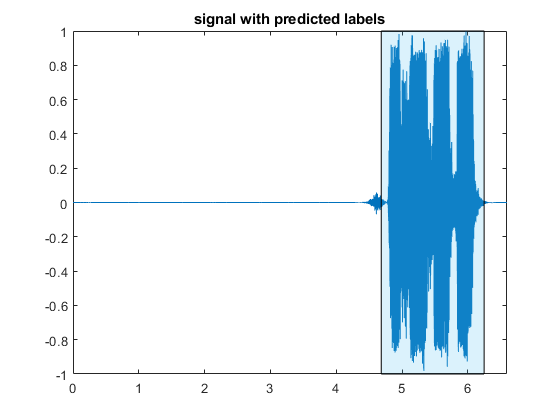
82



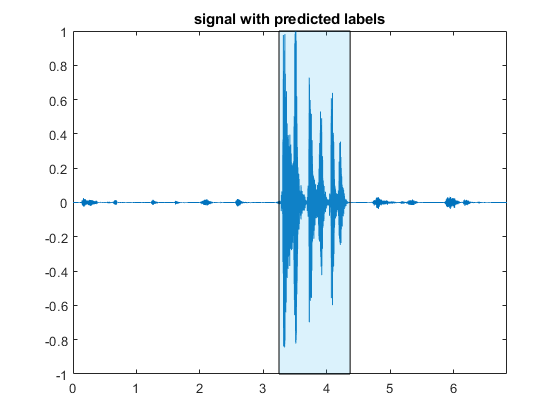
83



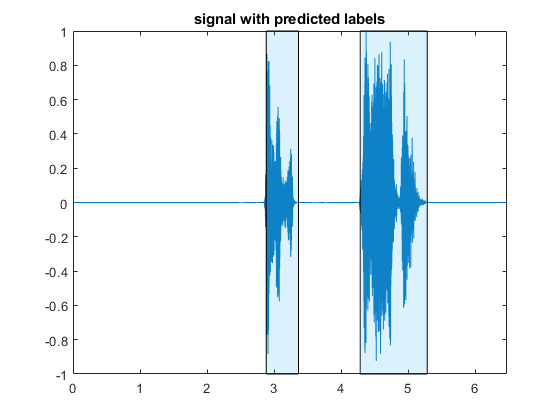
84



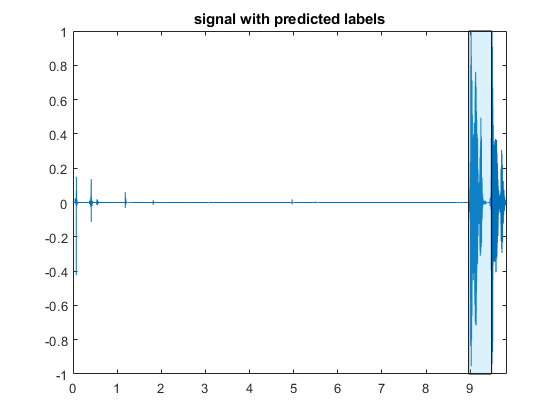
85



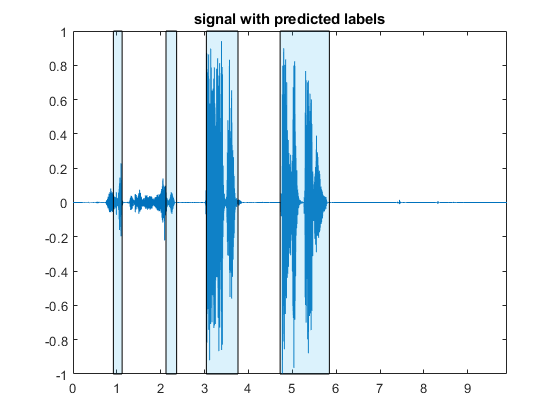
86



87



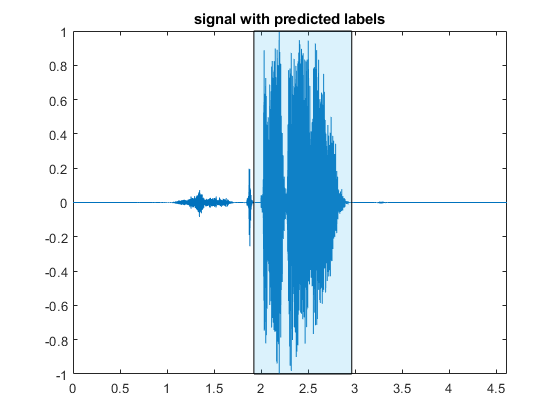
88



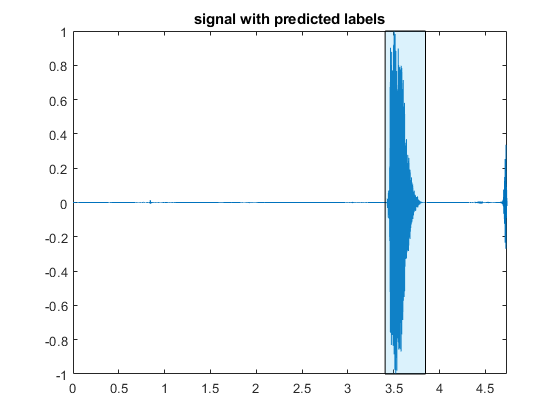
89



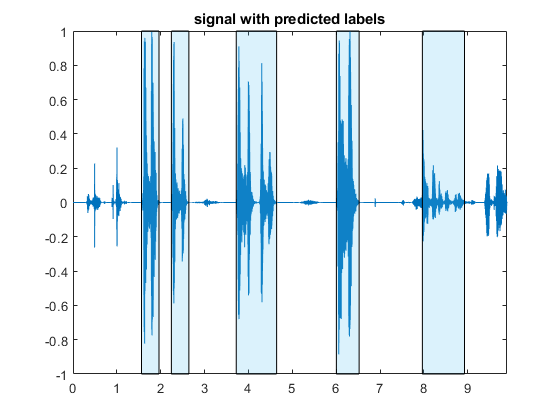
90



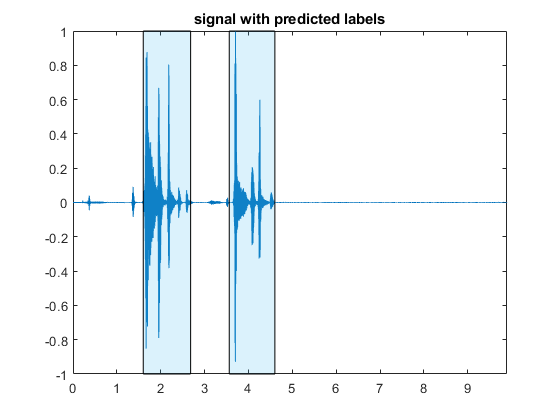
91



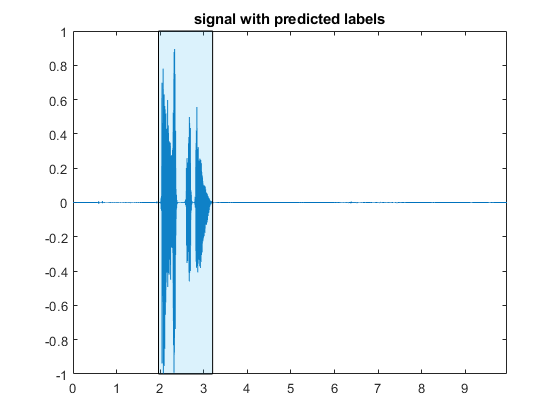
92



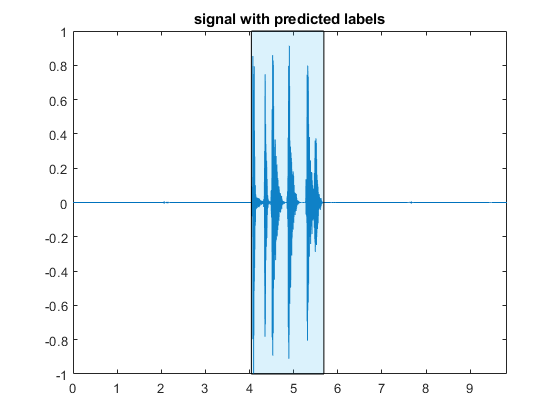
93



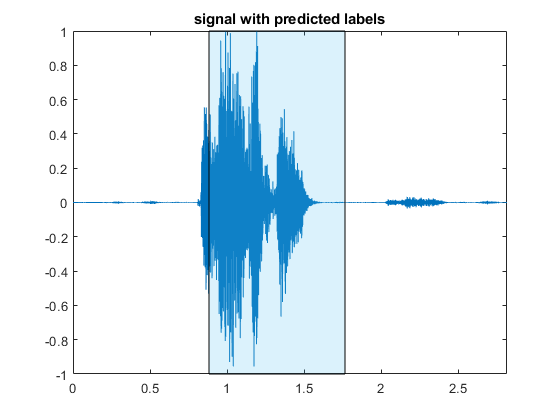
94



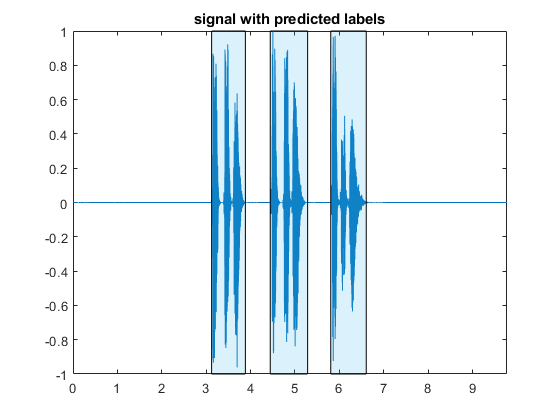
95



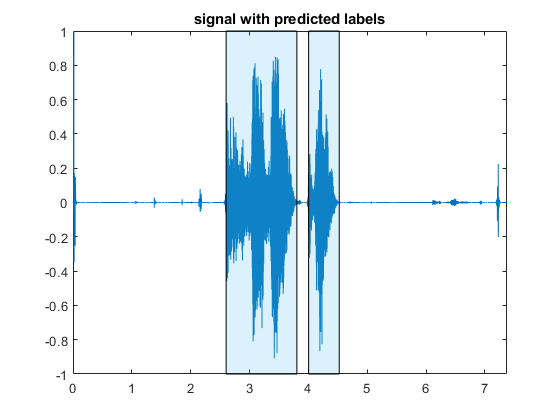
96



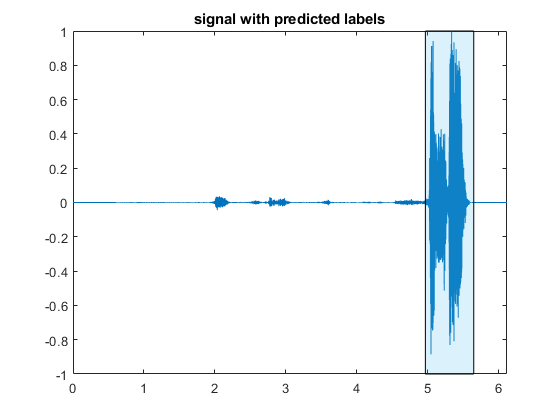
97



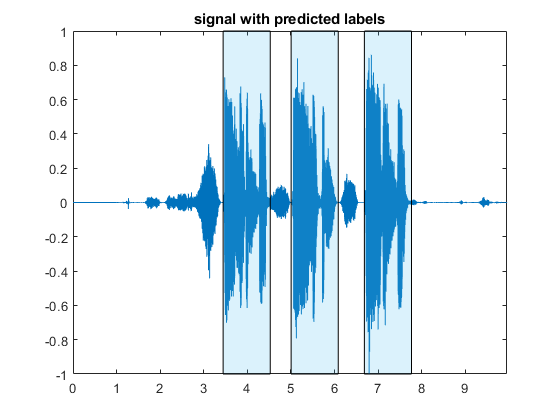
98



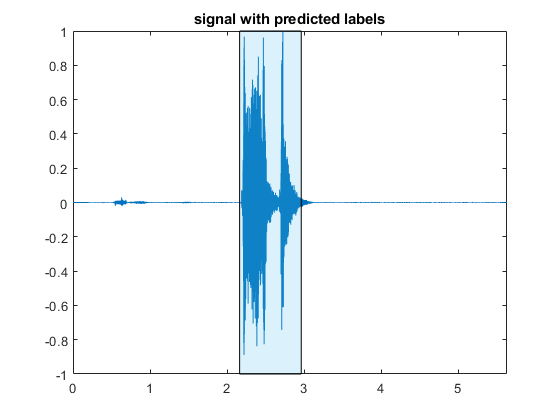
99



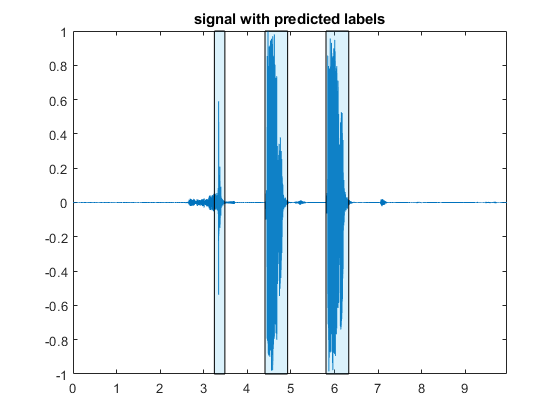
100



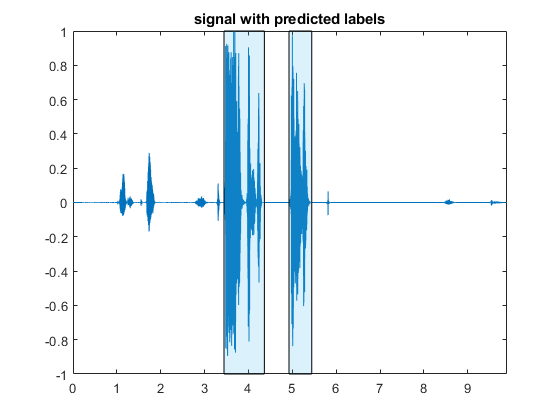
101



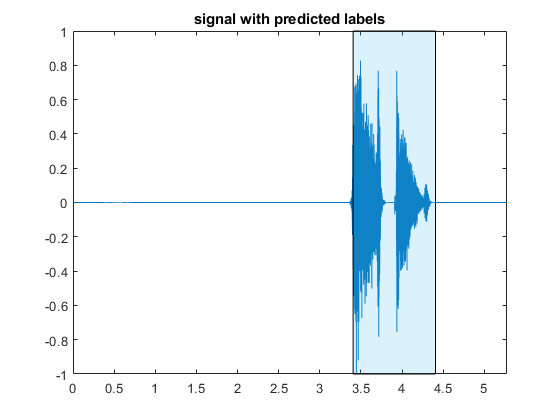
102



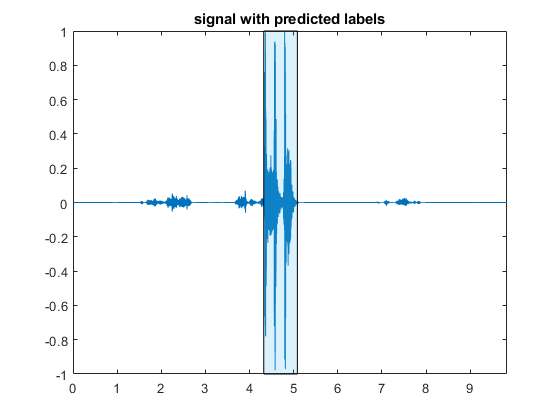
103



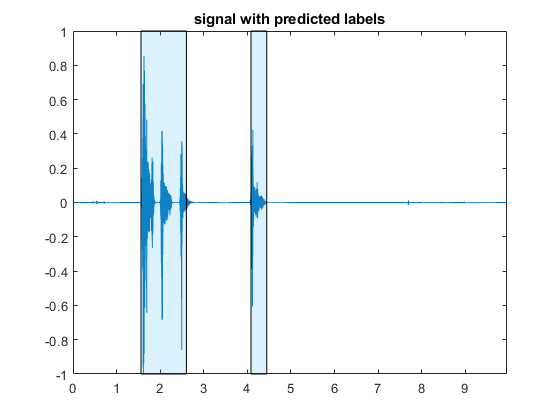
104



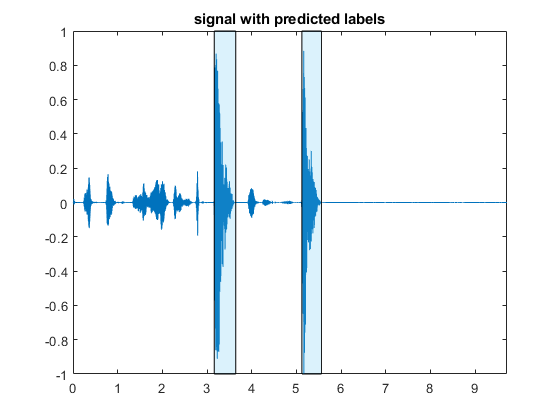
105



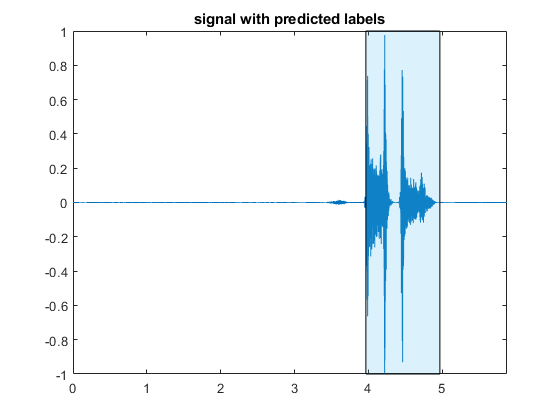
106



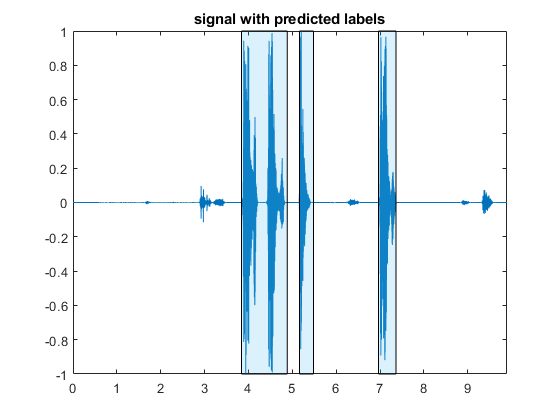
107



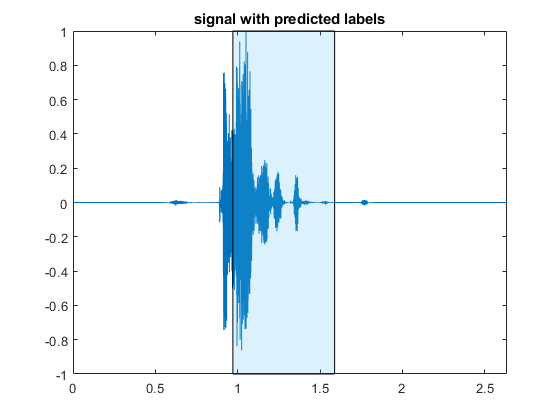
108



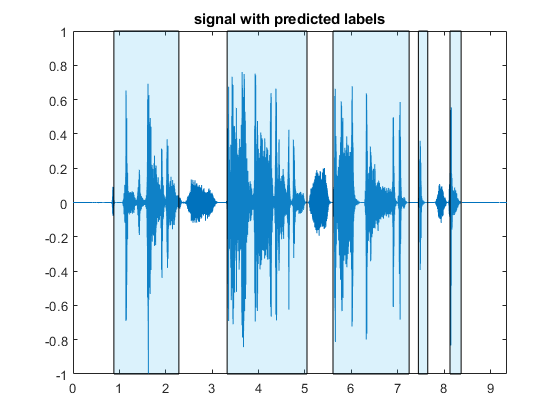
109



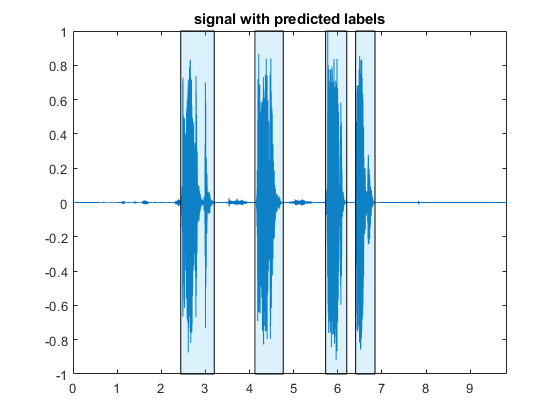
110



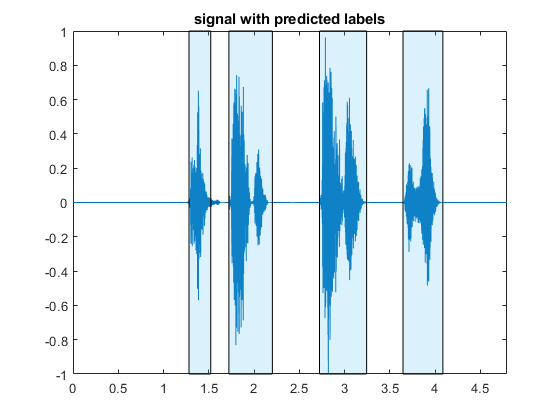
111



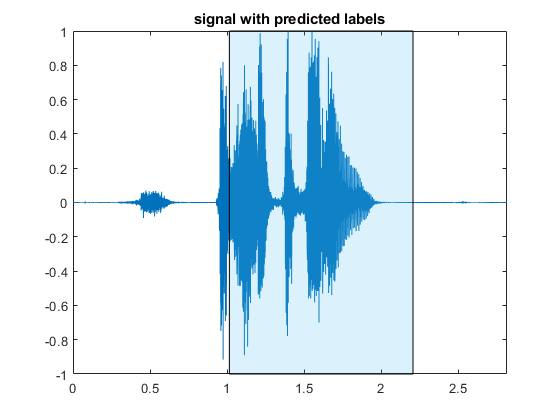
112



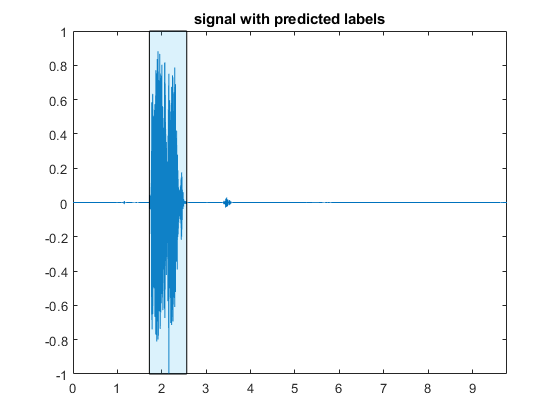
113



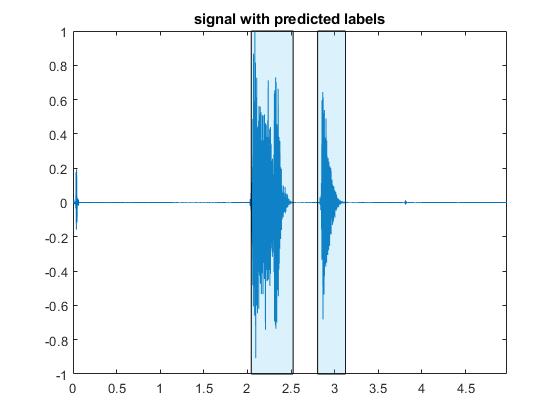
114



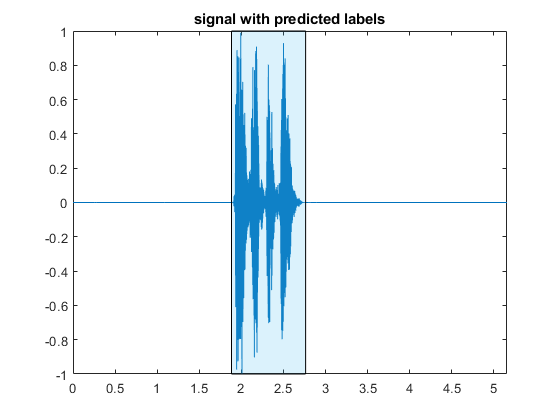
115



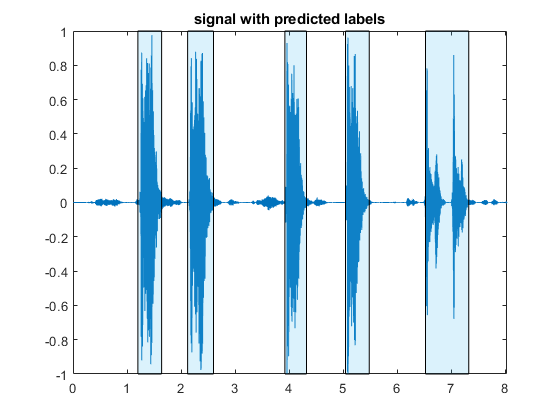
116



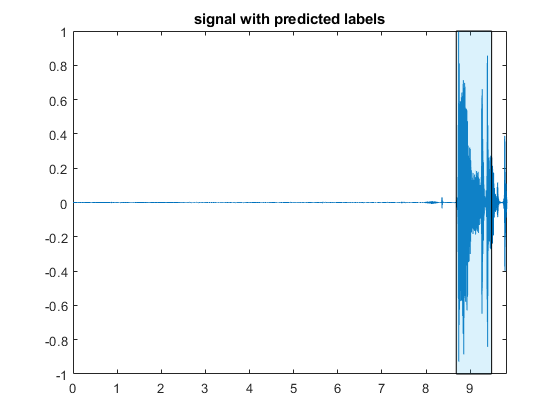
117



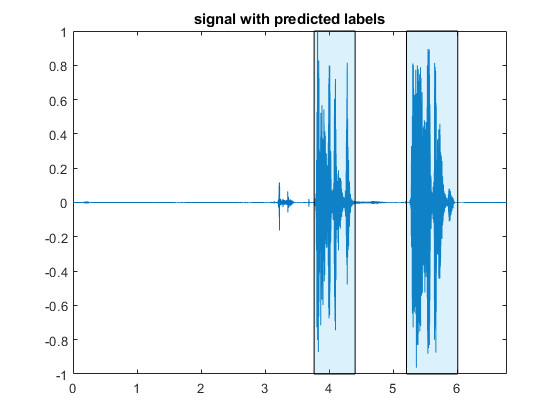
118



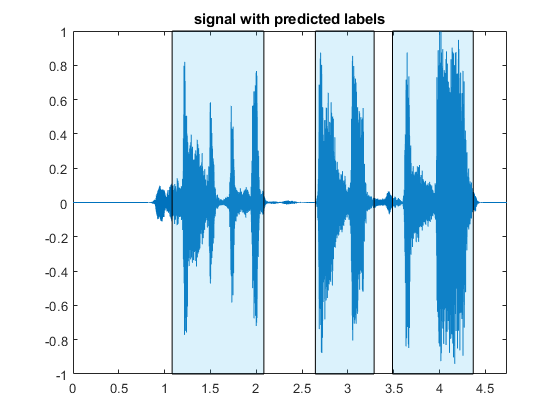
119



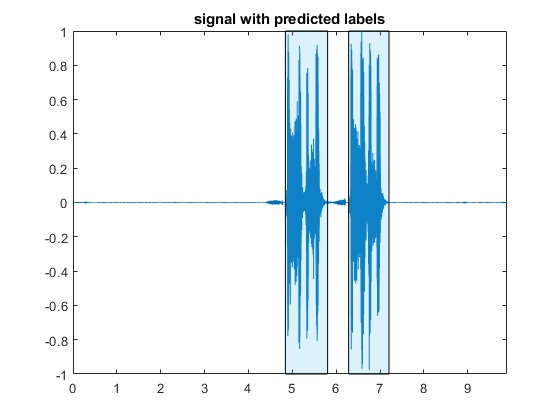
120



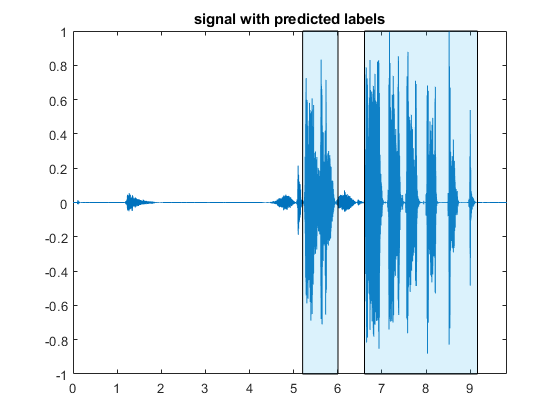
121



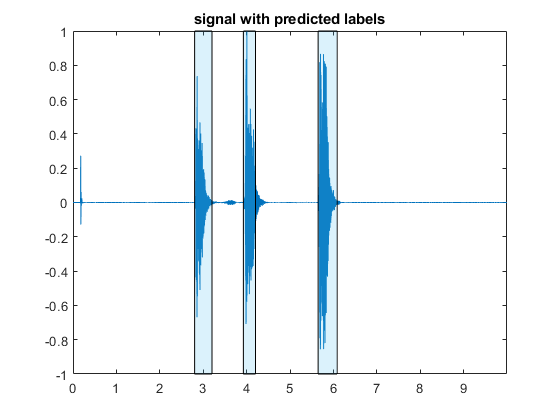
122



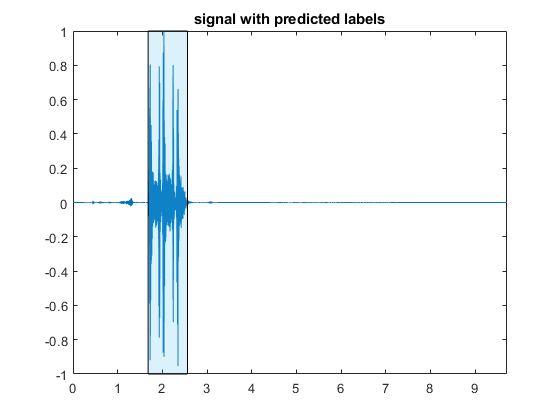
123



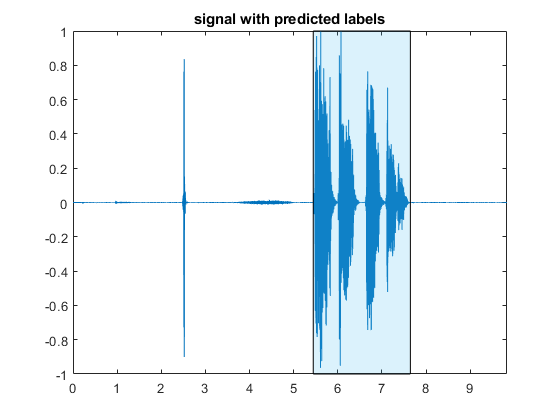
124



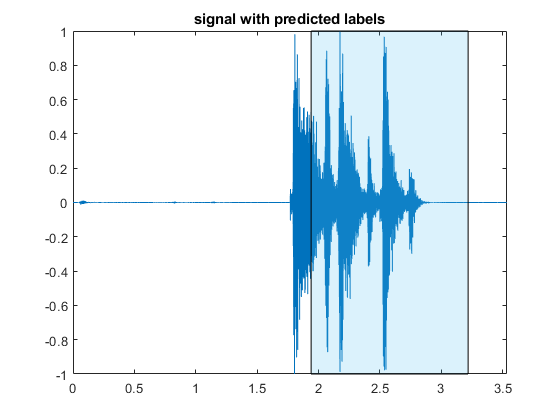
125



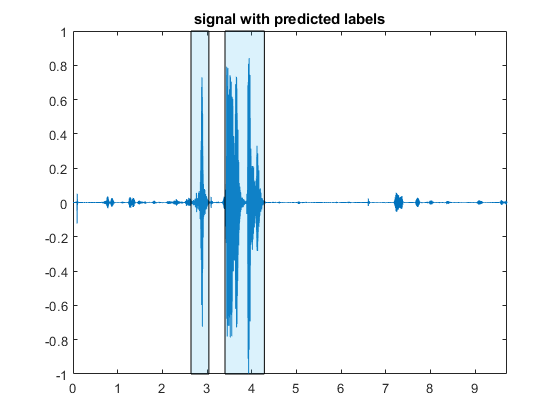
126



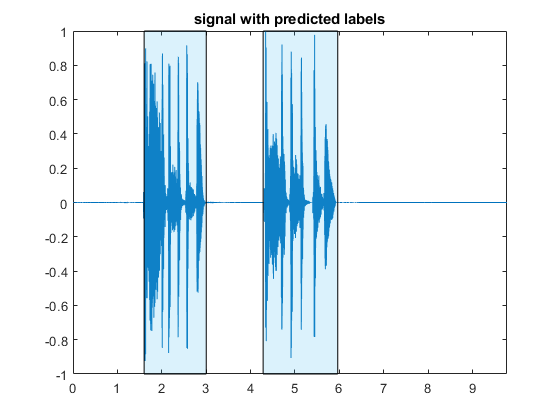
127



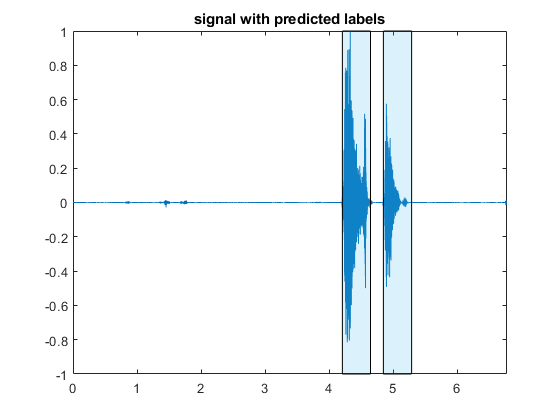
128



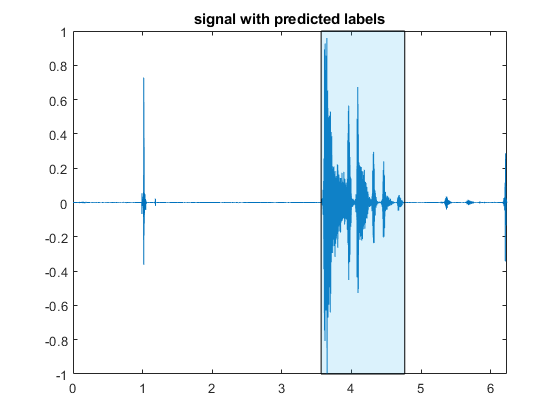
129



130



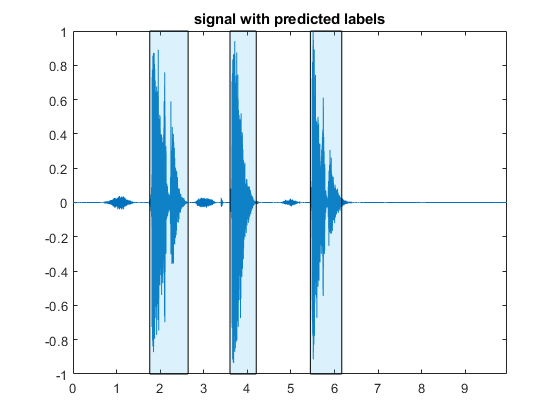
131



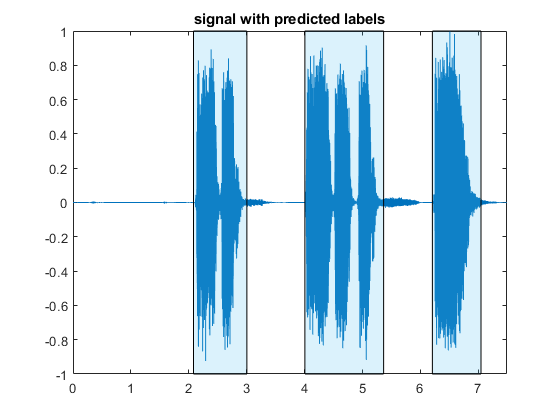
132



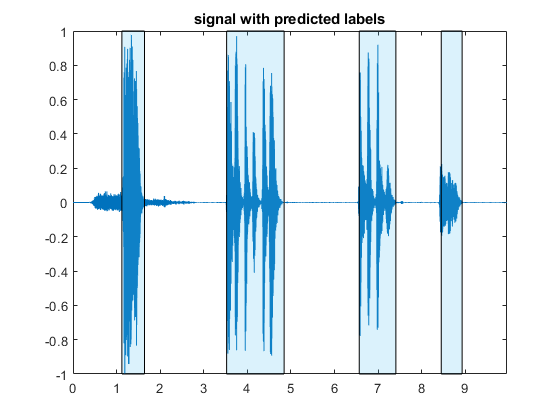
133



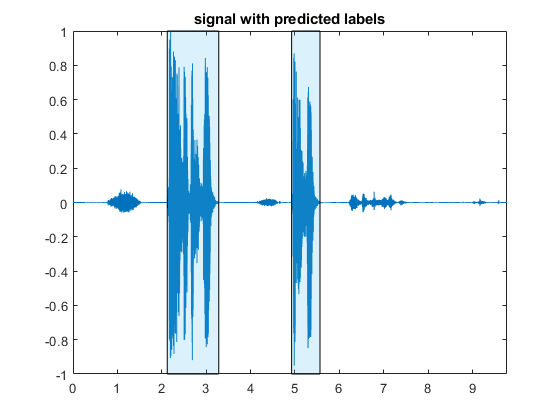
134



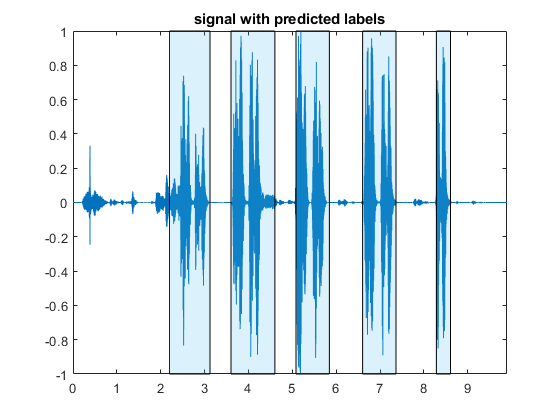
135



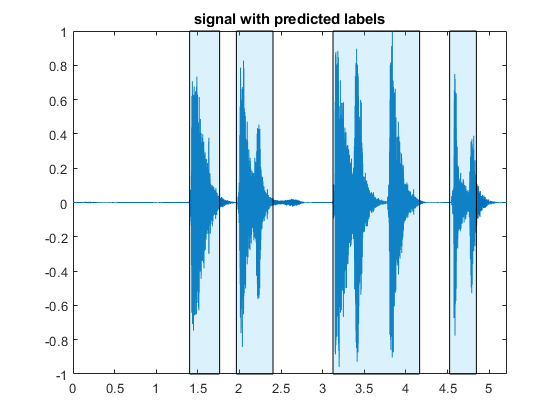
136



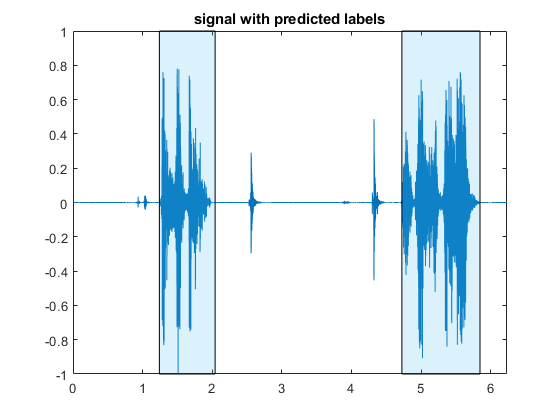
137



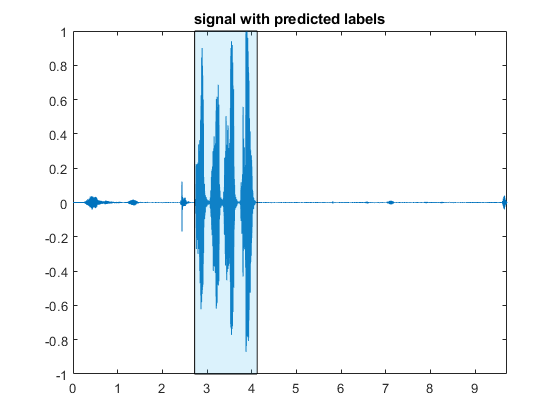
138



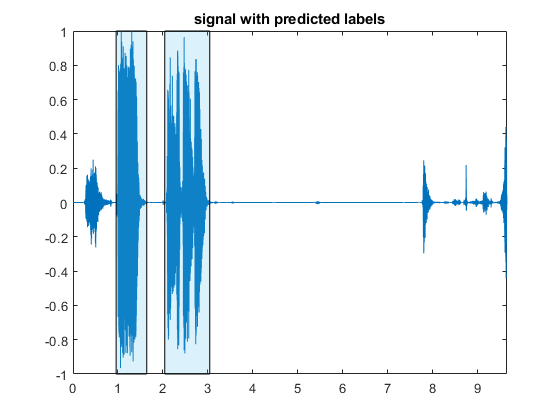
139



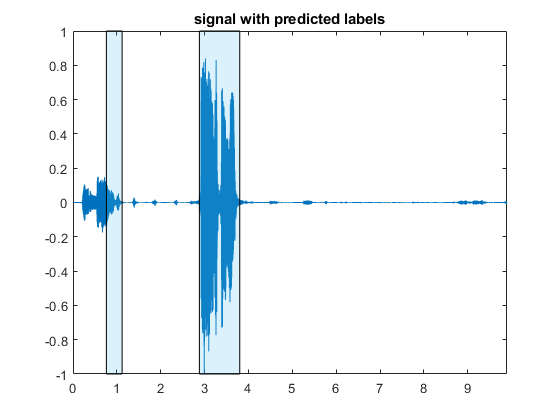
140



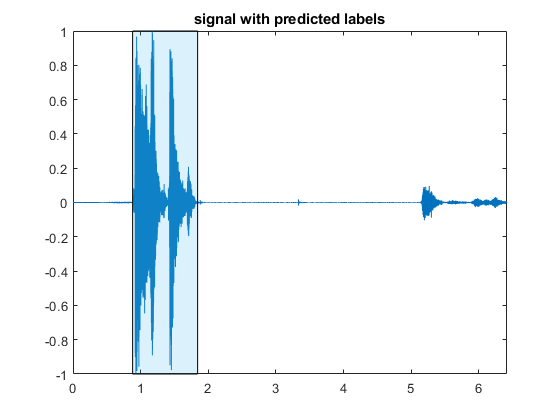
141



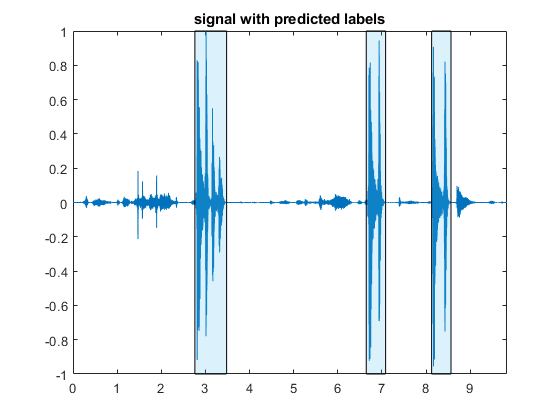
142



143



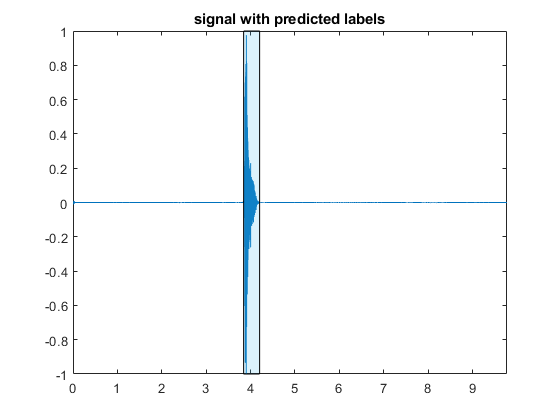
144



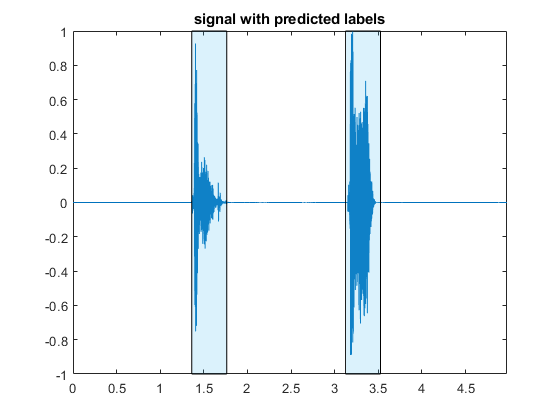
145



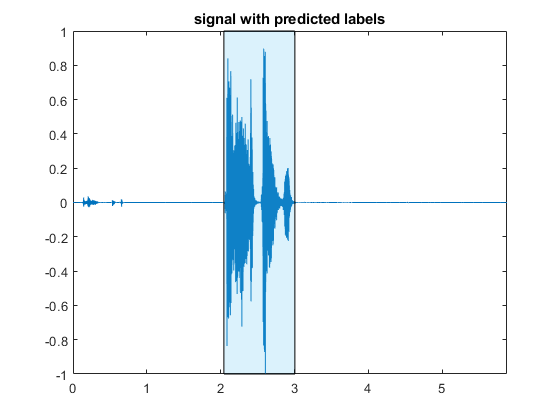
146



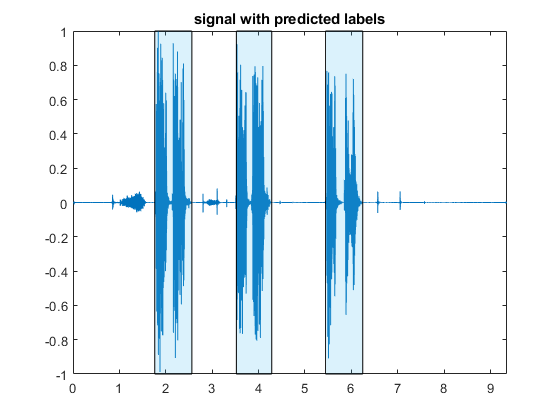
147



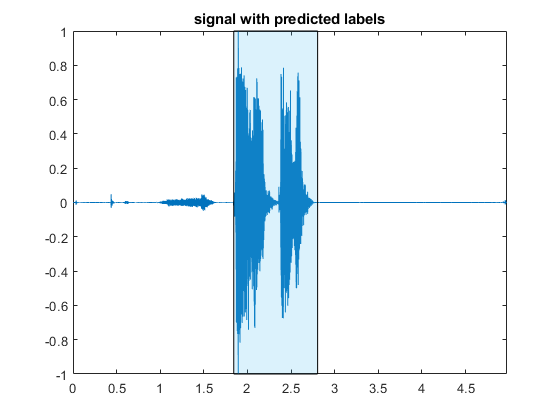
148



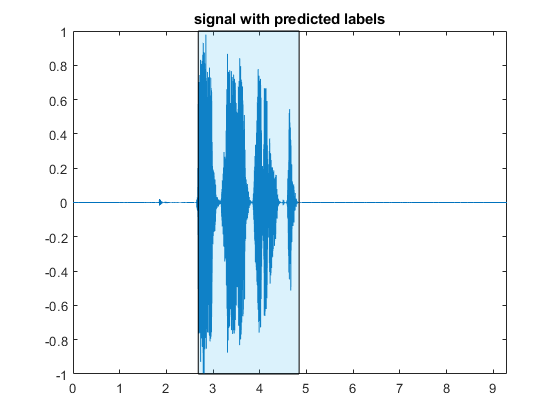
149



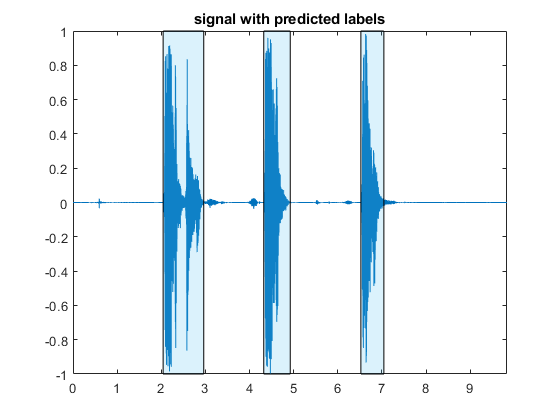
150



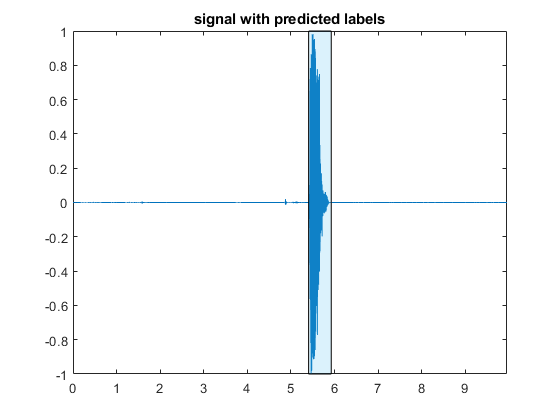
151



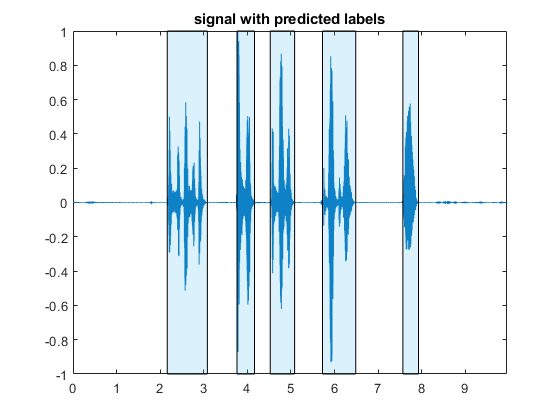
152



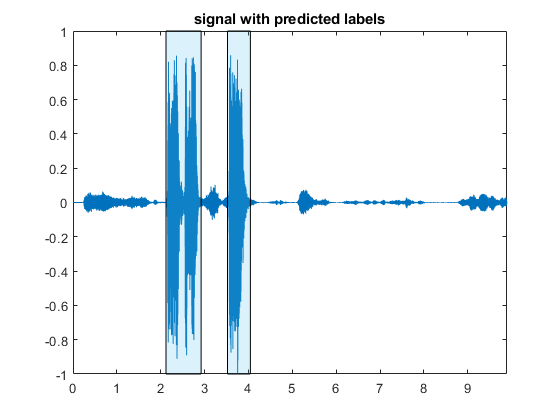
153



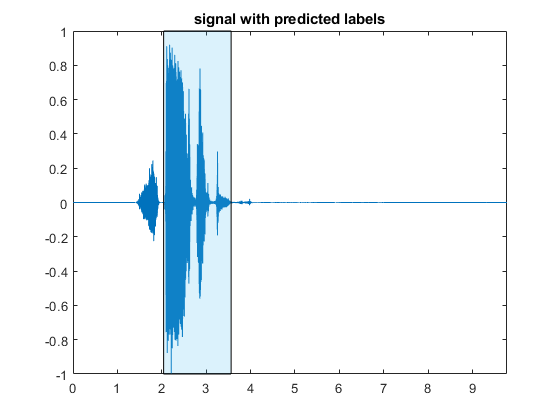
154



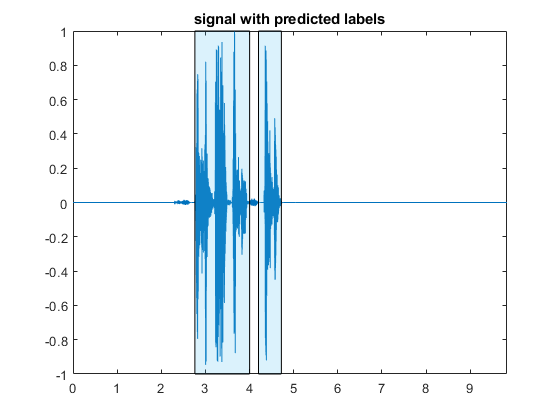
155



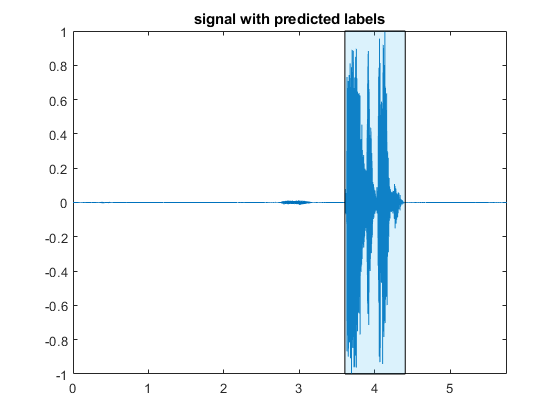
156



157



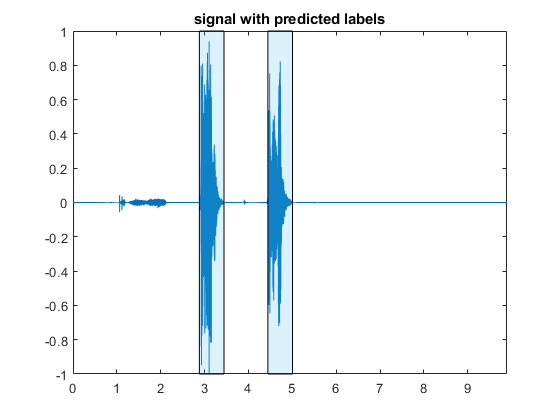
158



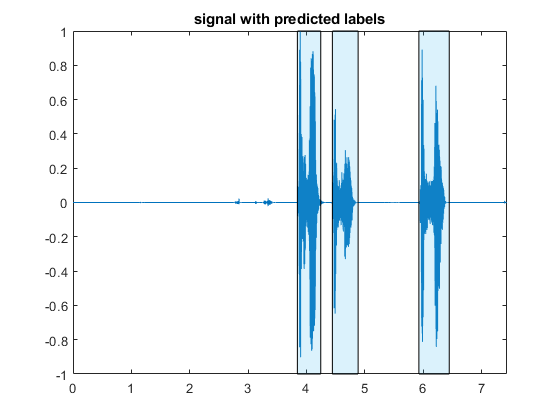
159



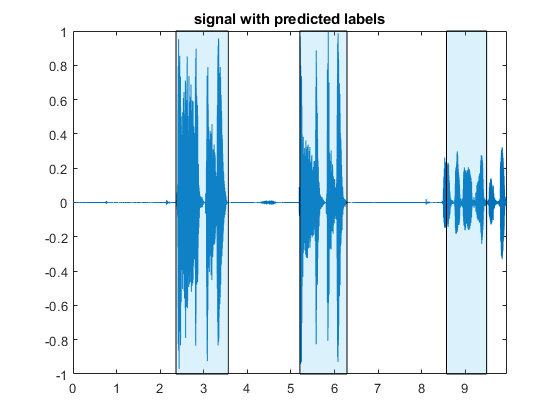
160



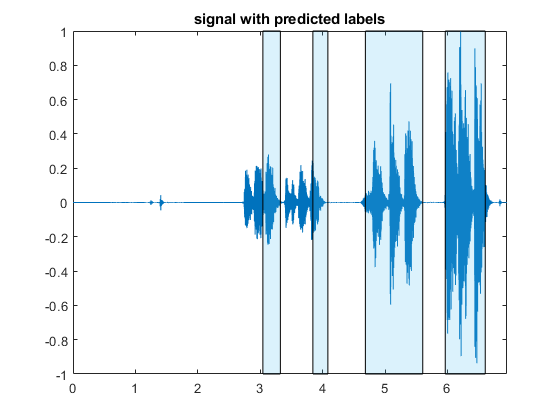
161



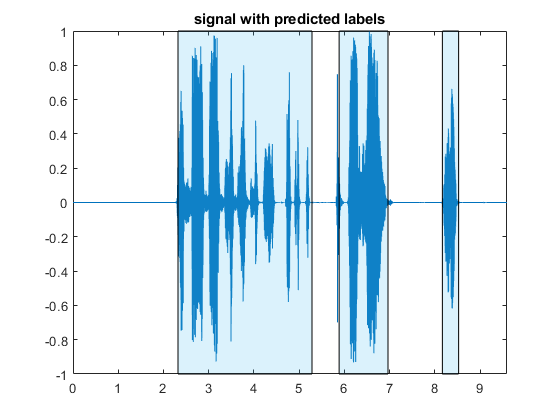
162



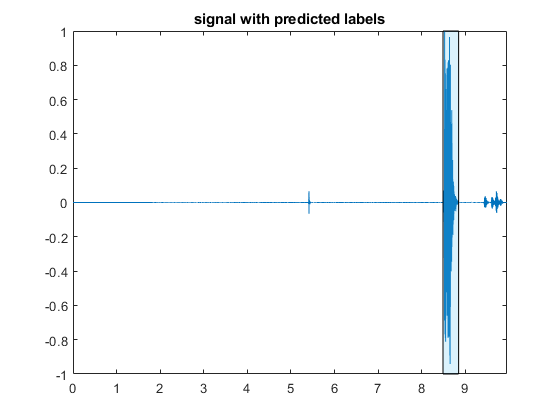
163



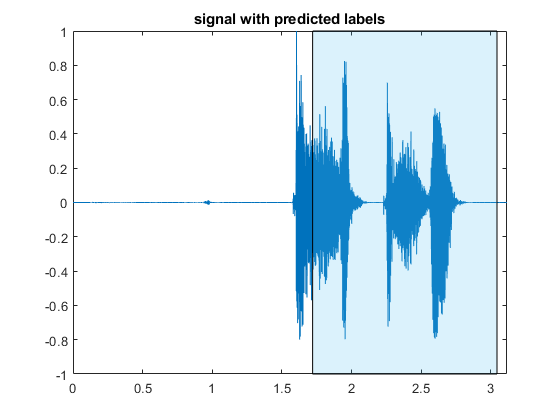
164



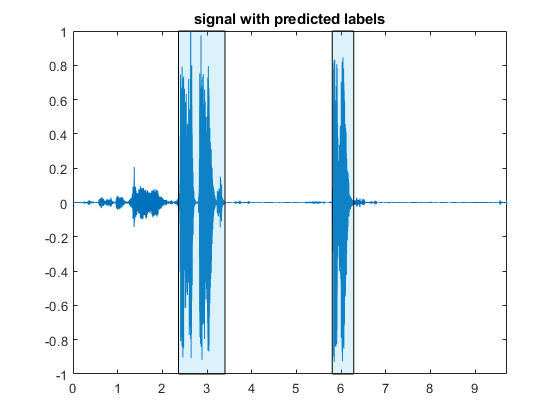
165



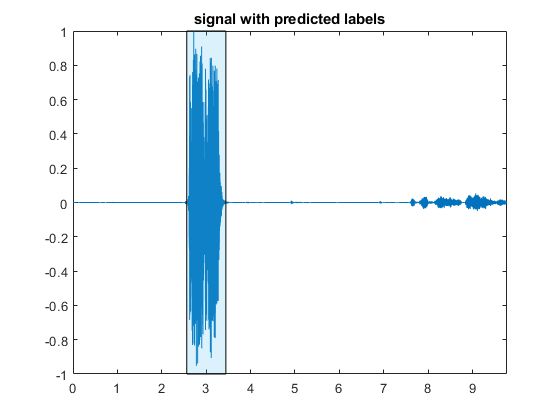
166



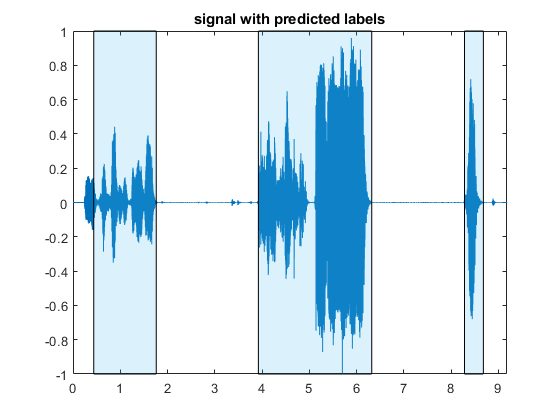
167



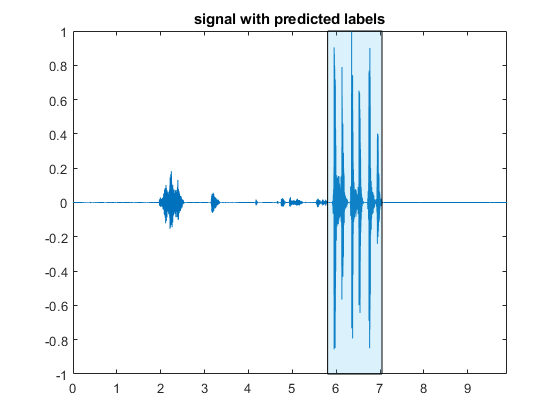
168



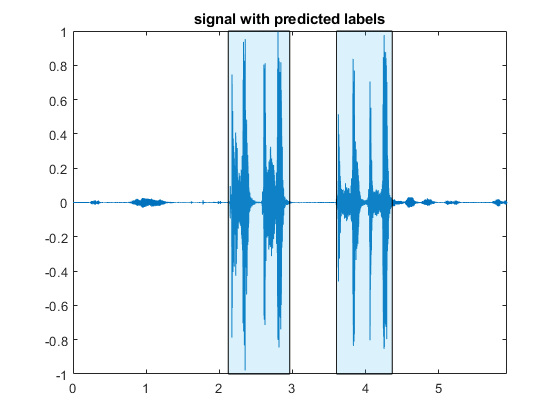
169



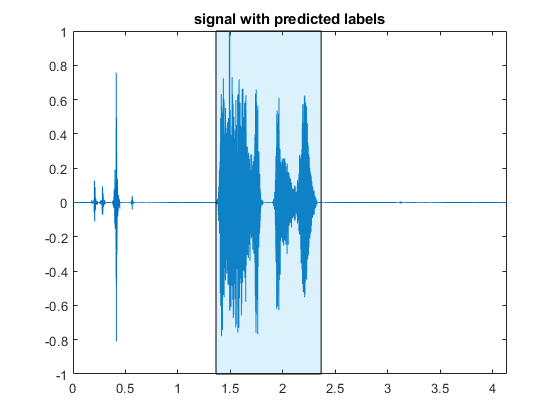
170



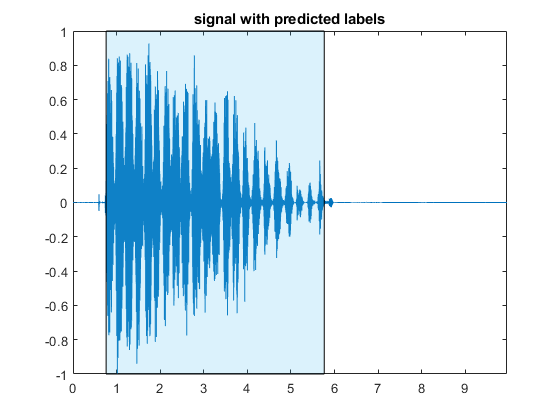
171



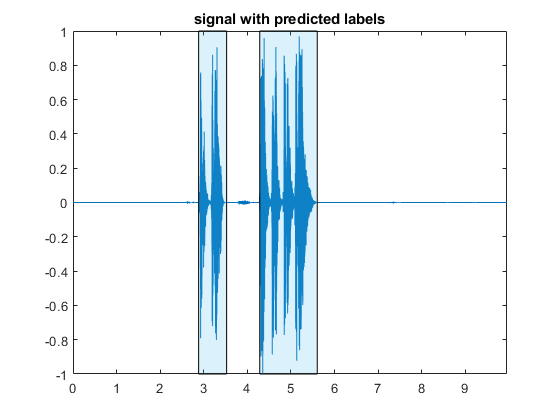
172



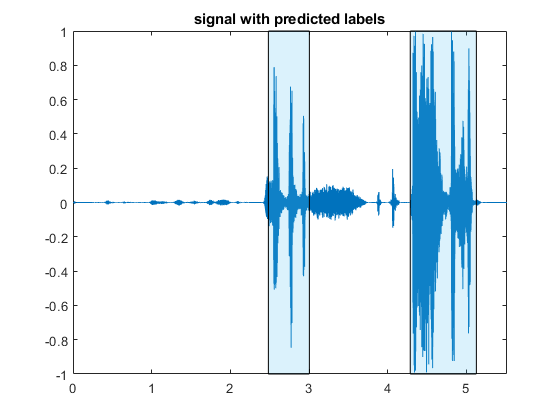
173



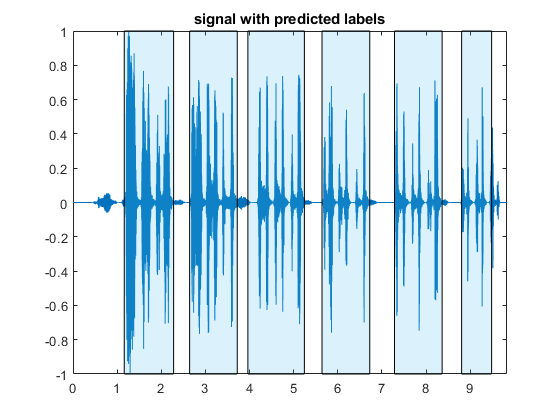
174



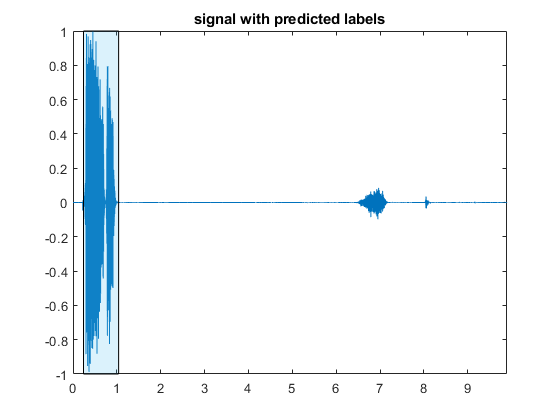
175



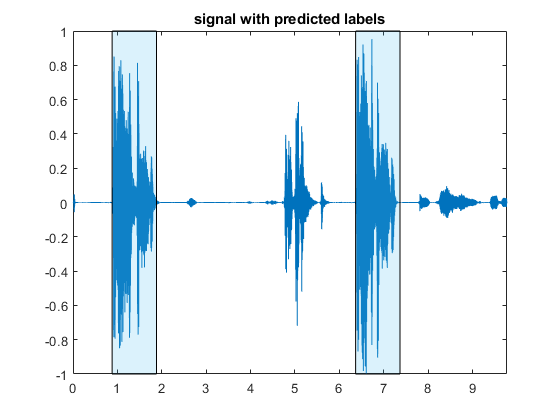
176



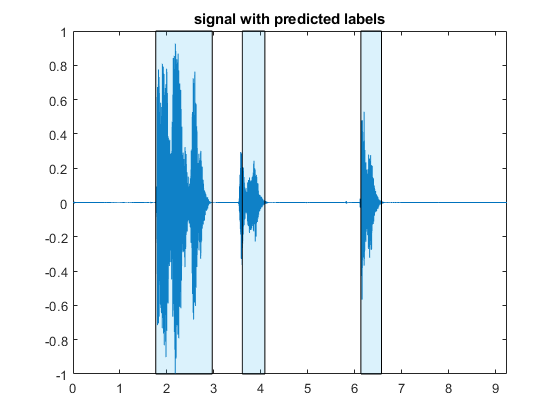
177



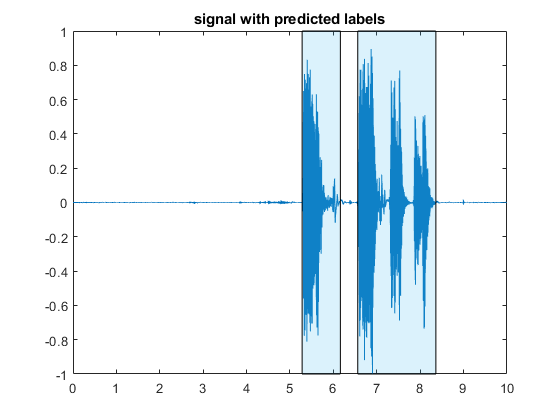
178



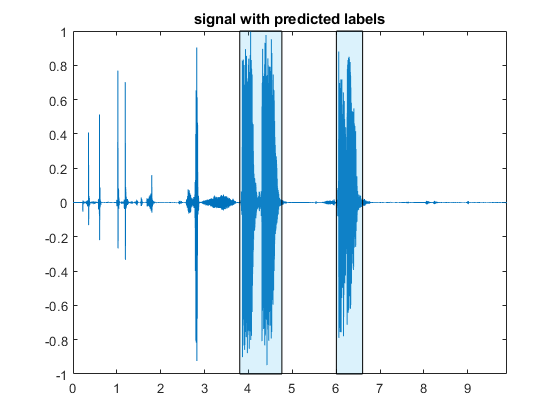
179



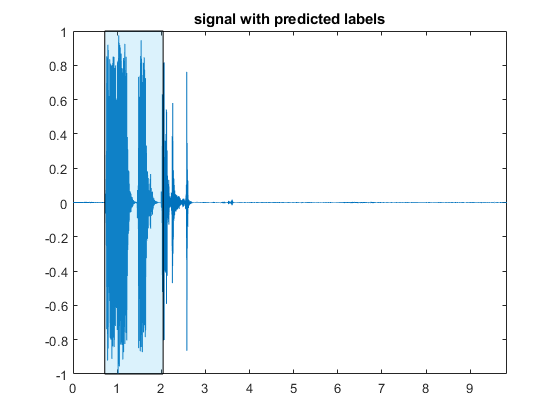
180



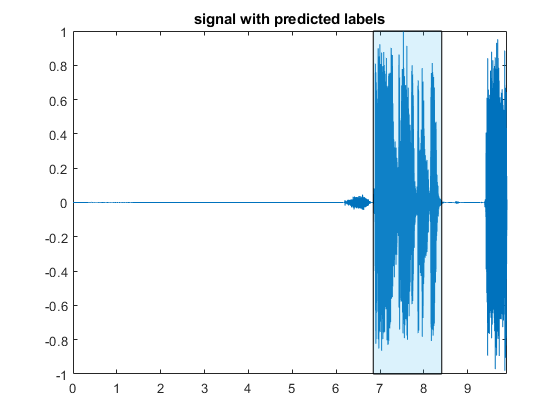
181



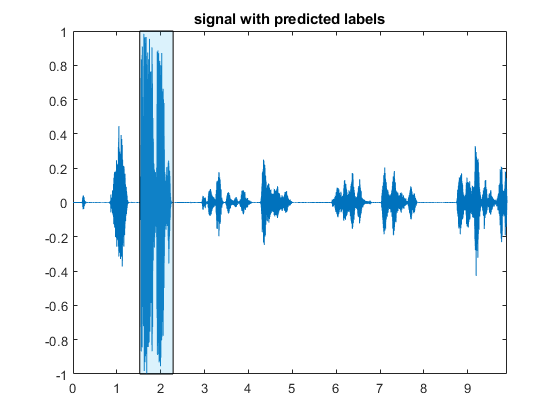
182



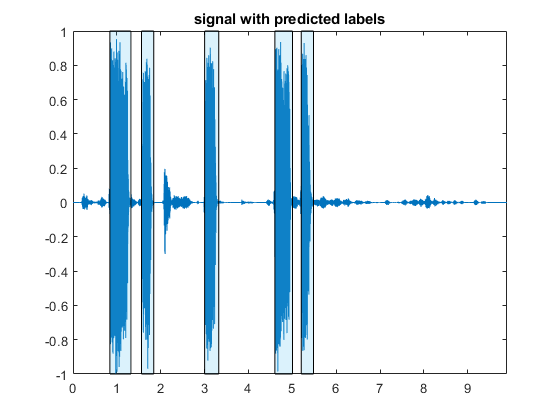
183



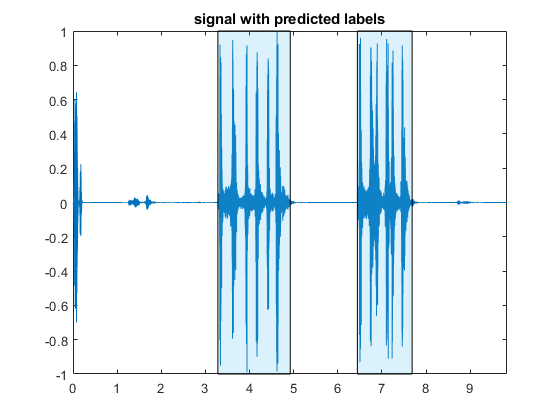
184



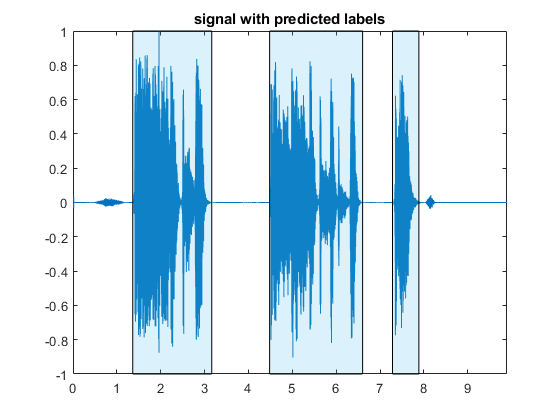
185



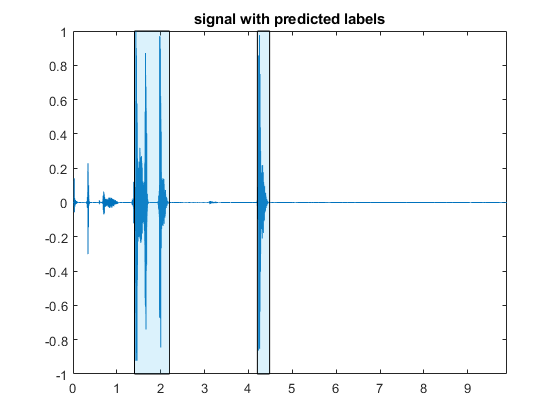
186



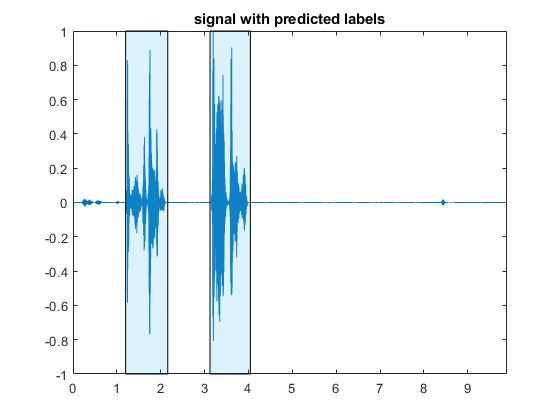
187



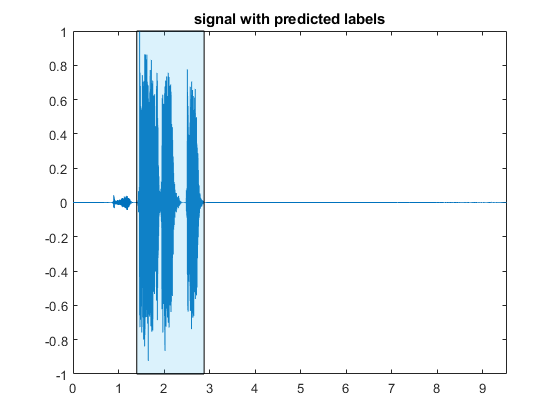
188



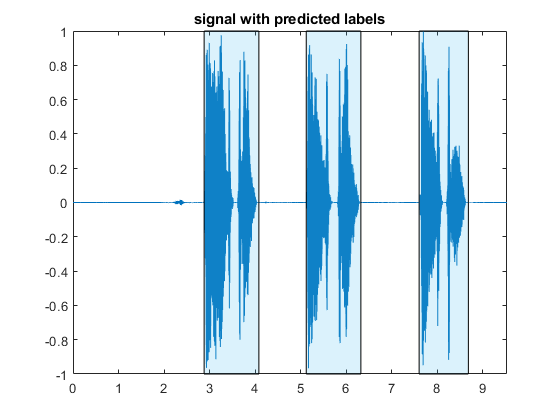
189



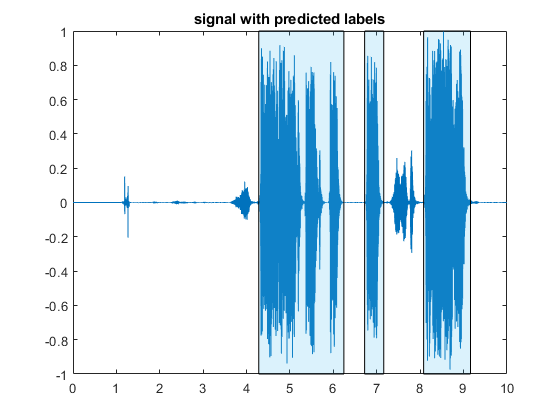
190



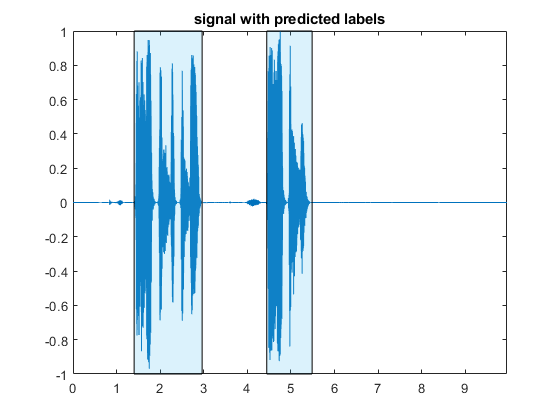
191



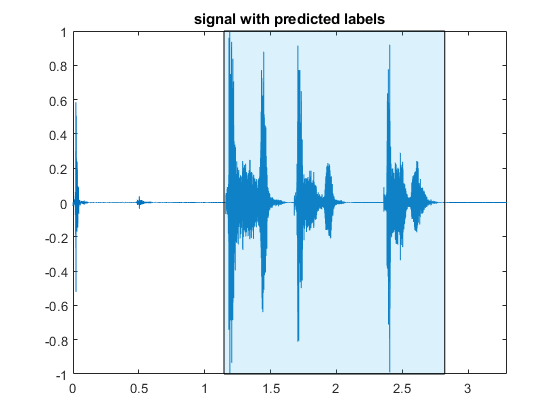
192



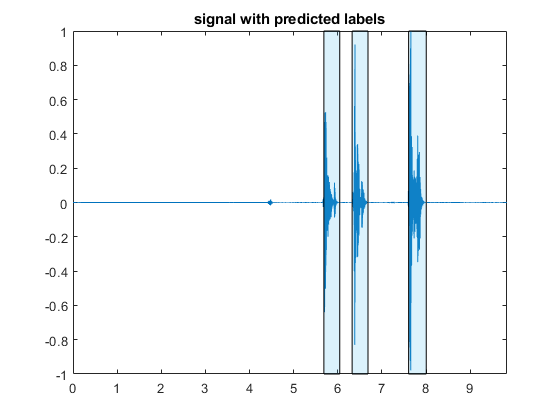
193



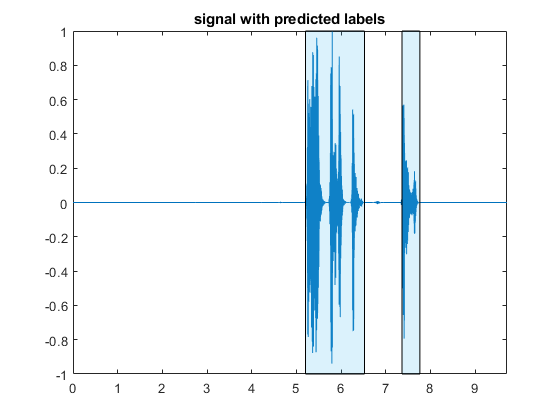
194



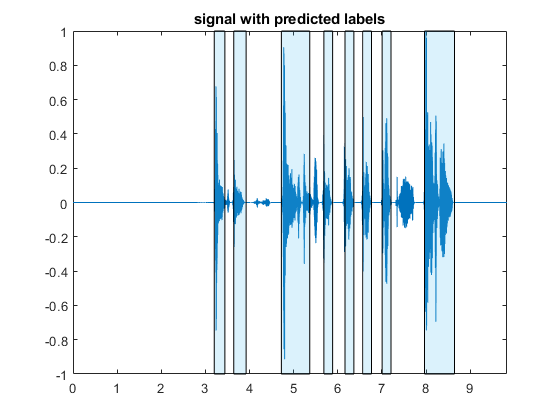
195



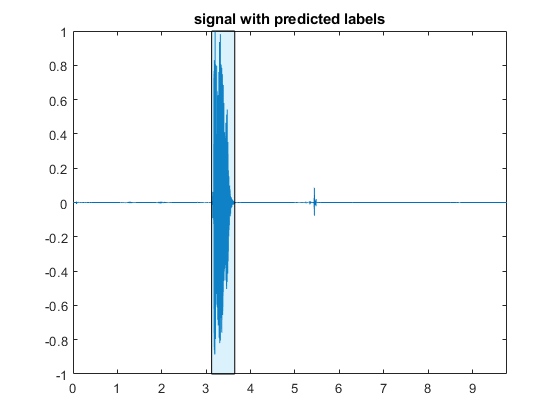
196



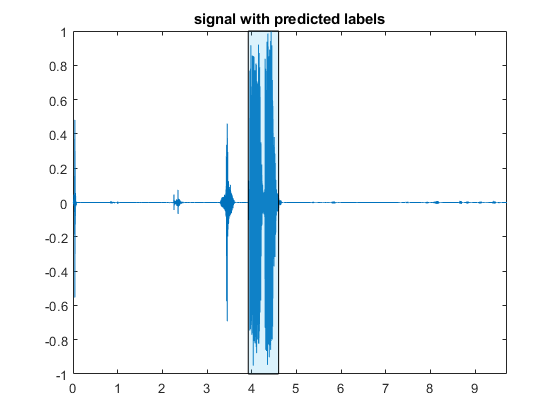
197



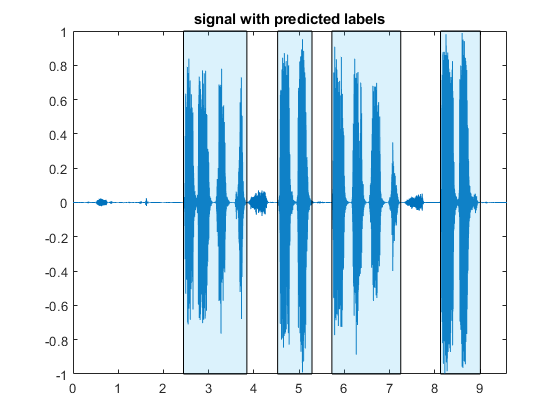
198



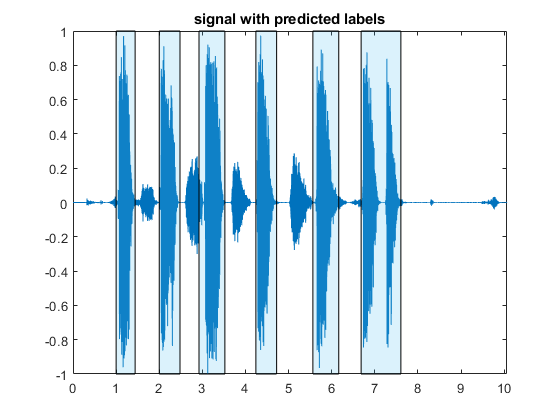
199



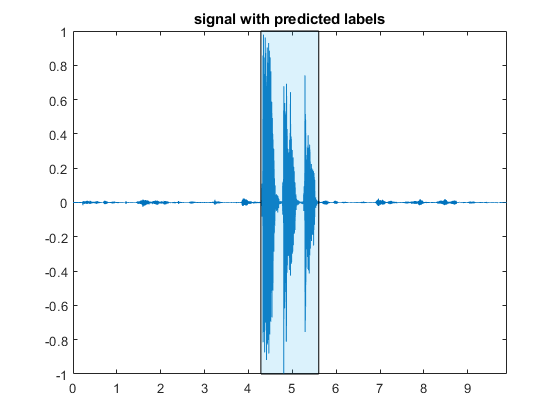
200



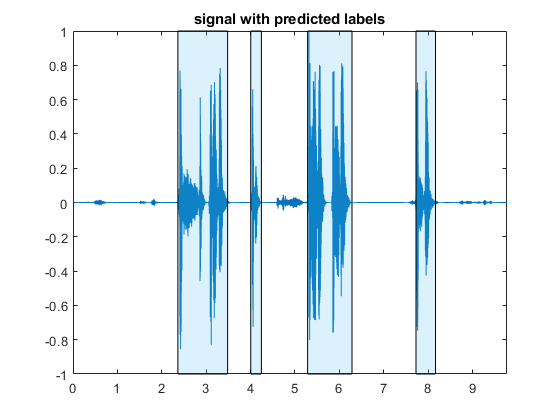
201



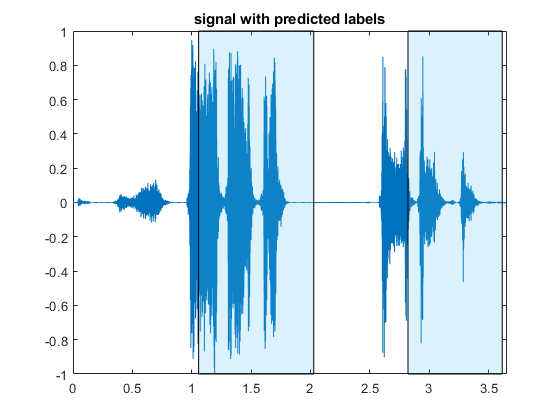
202



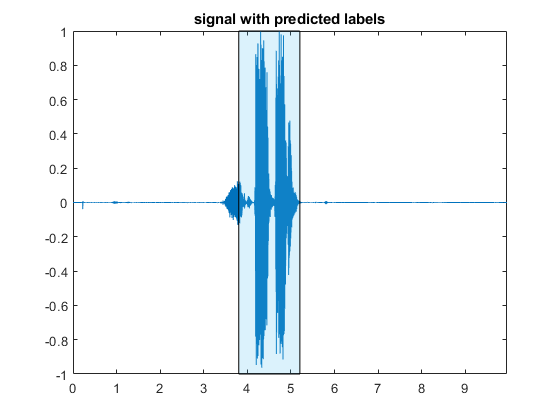
203



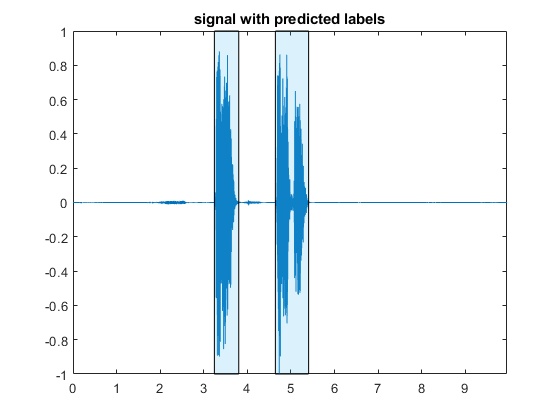
204



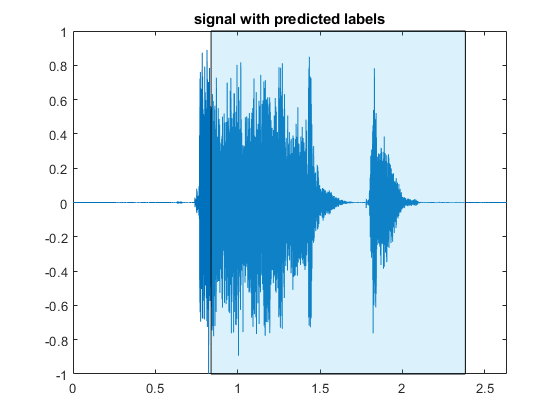
205



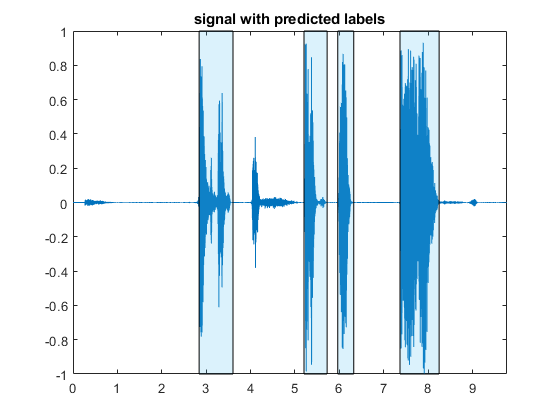
206



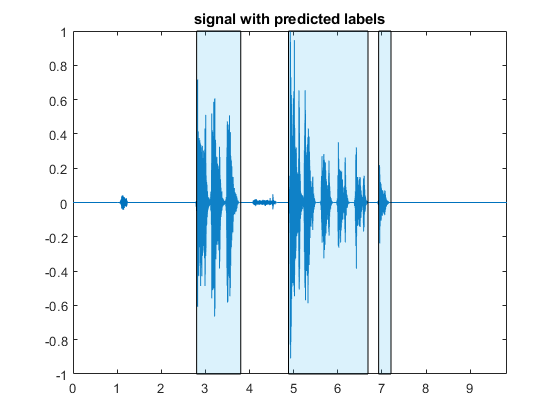
207



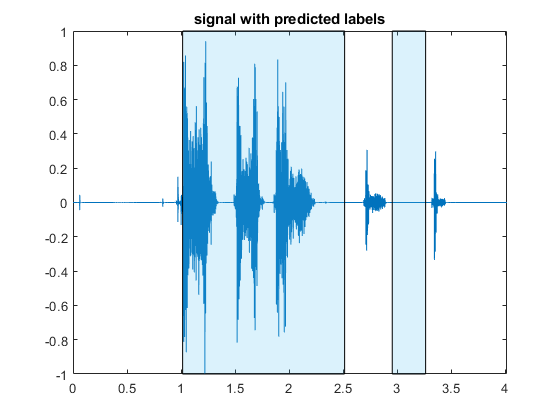
208



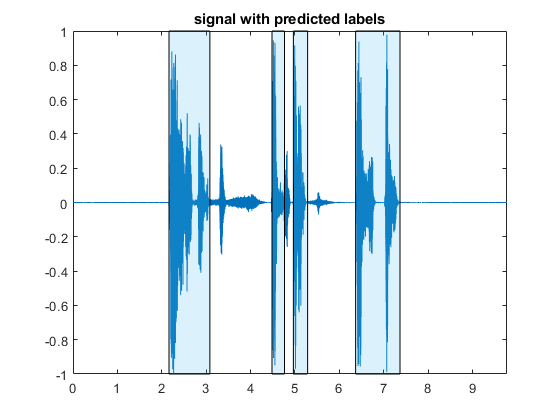
209



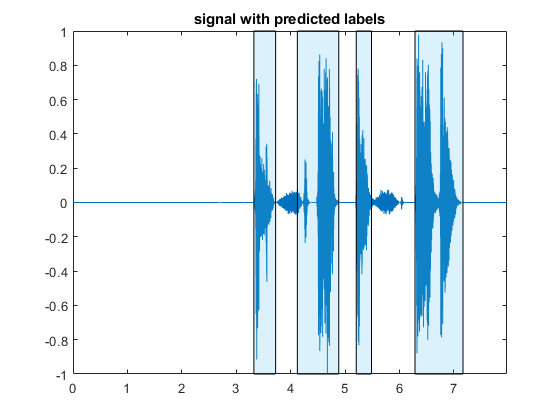
210



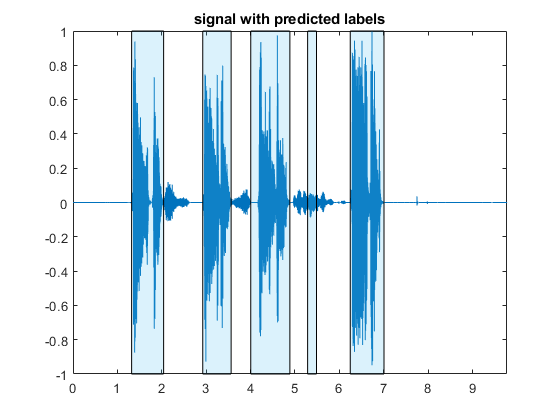
211



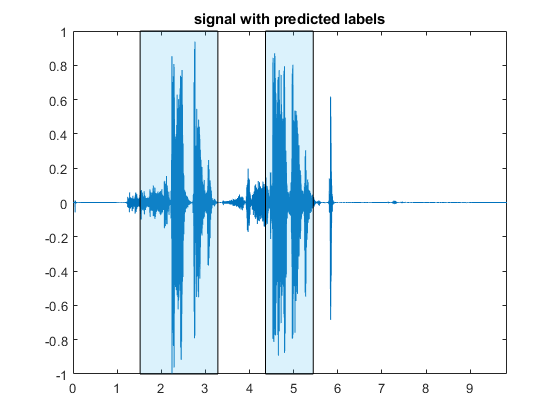
212



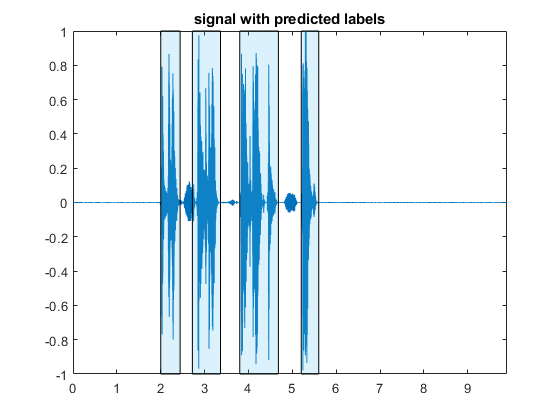
213



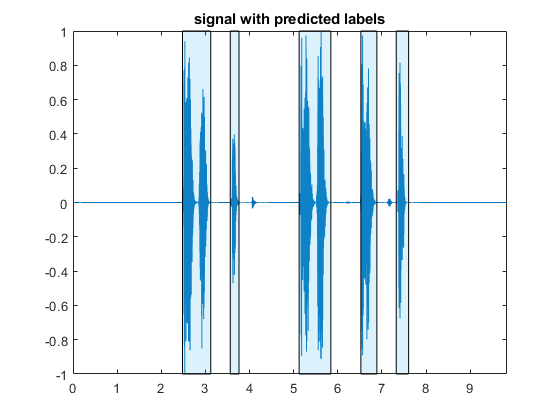
214



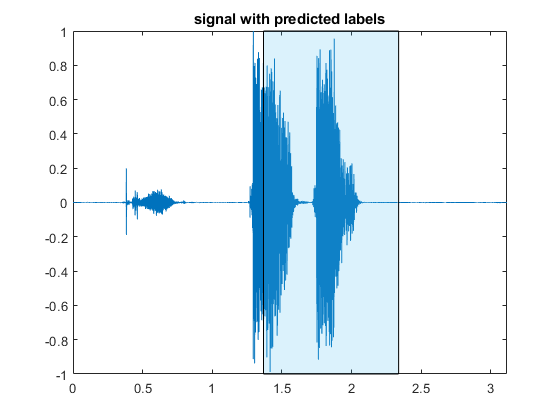
215



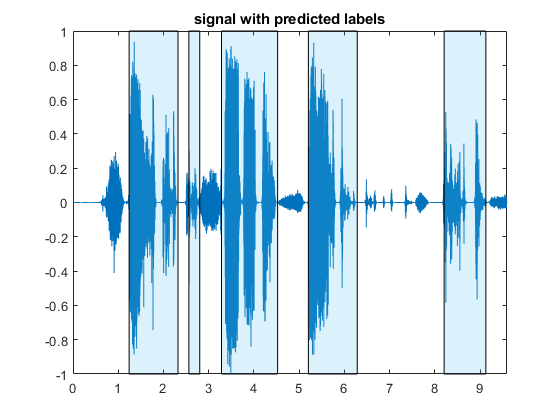
216



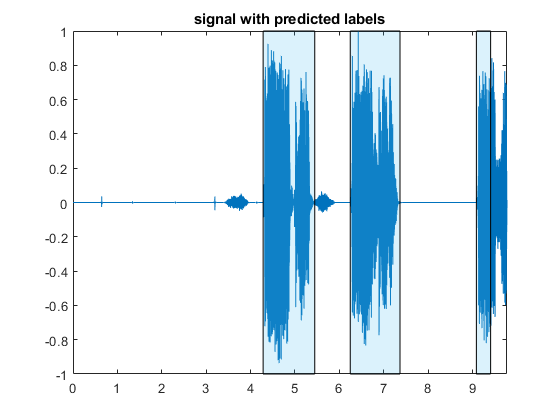
217



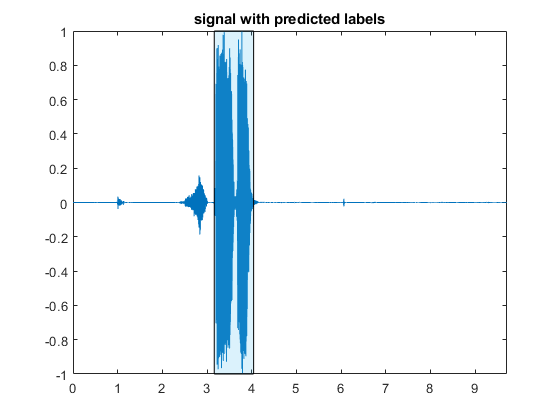
218



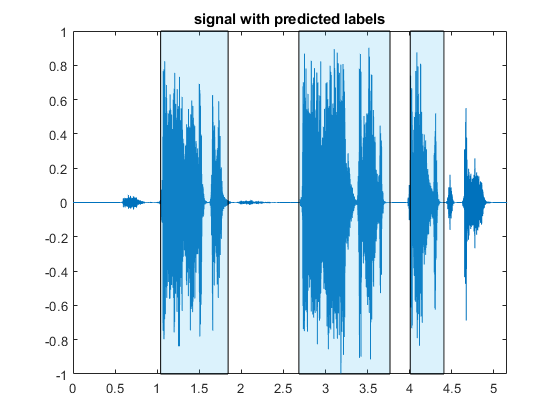
219



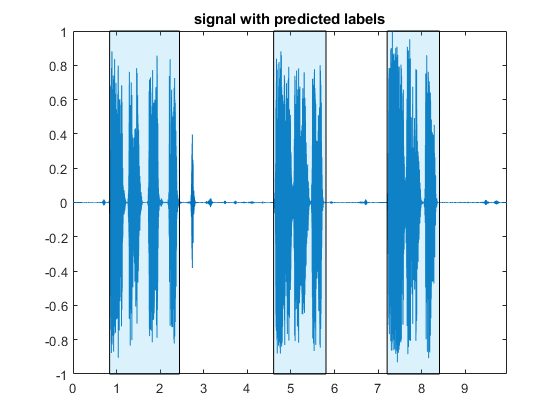
220



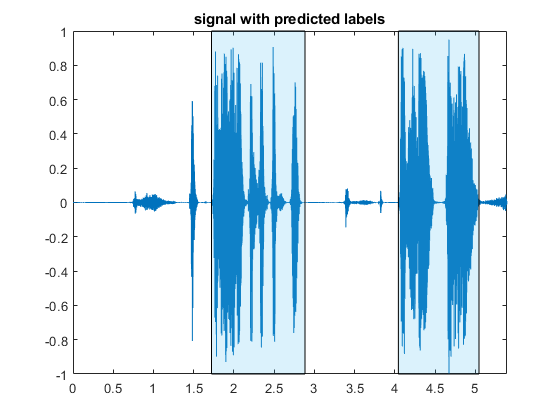
221



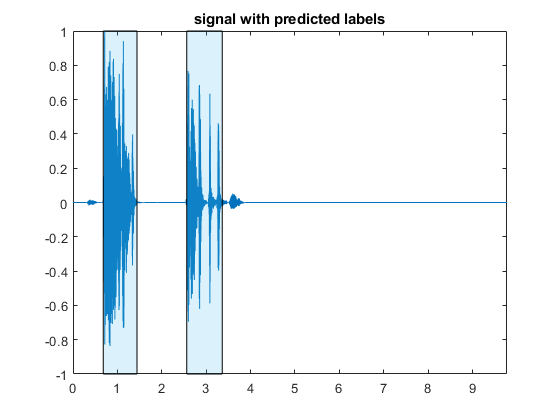
222



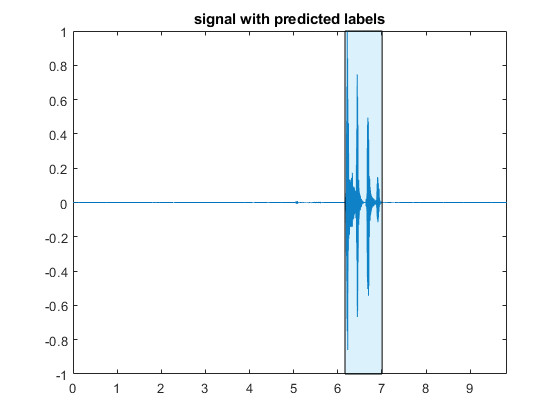
223



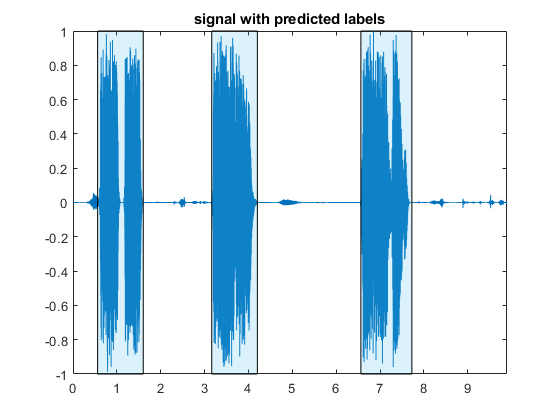
224



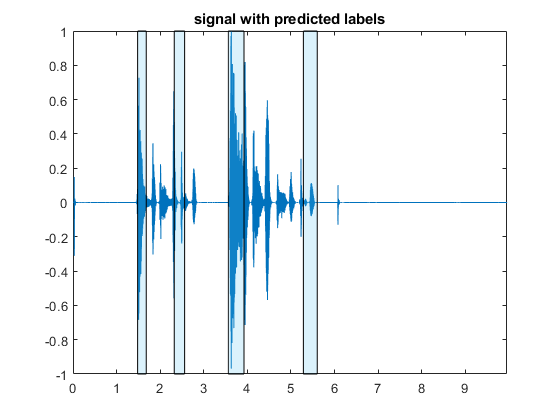
225



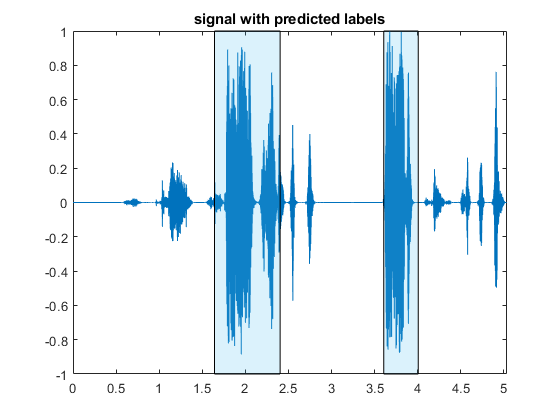
226



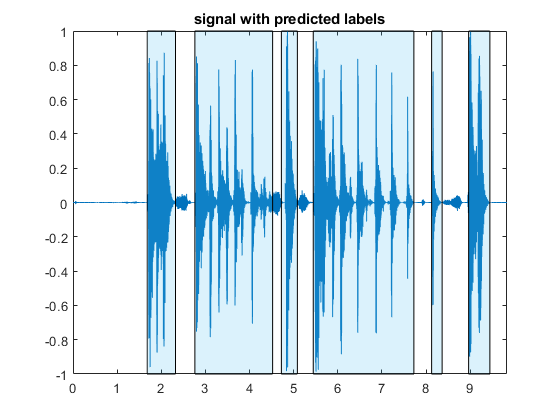
227



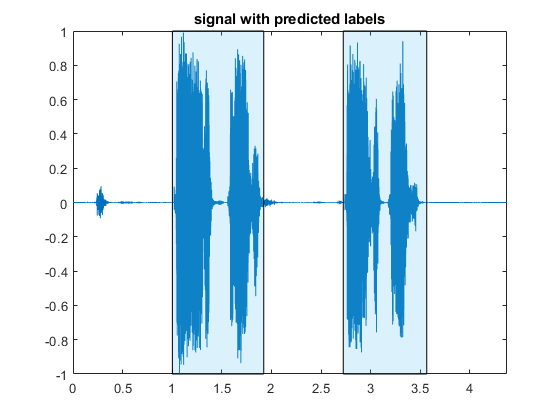
228



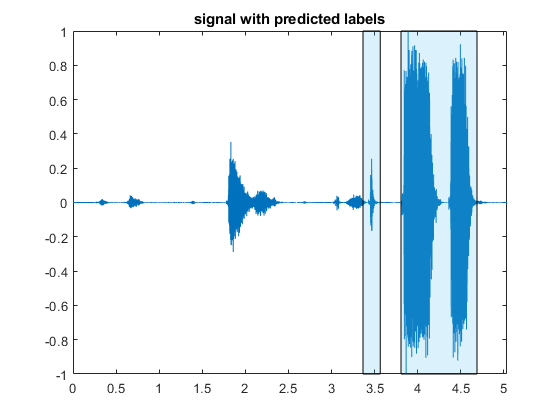
229



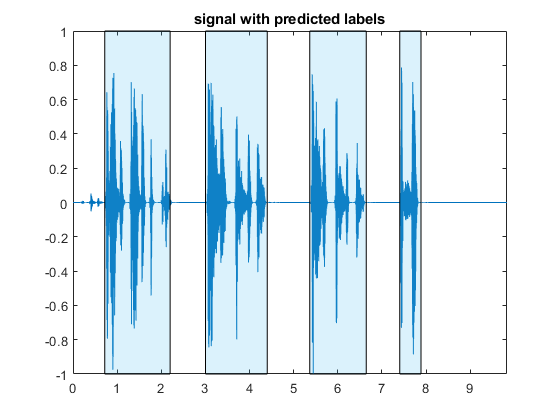
230



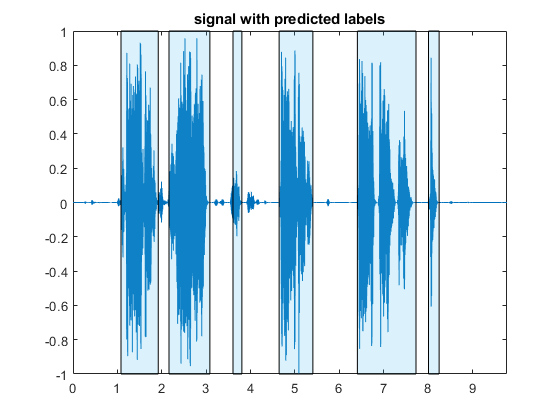
231



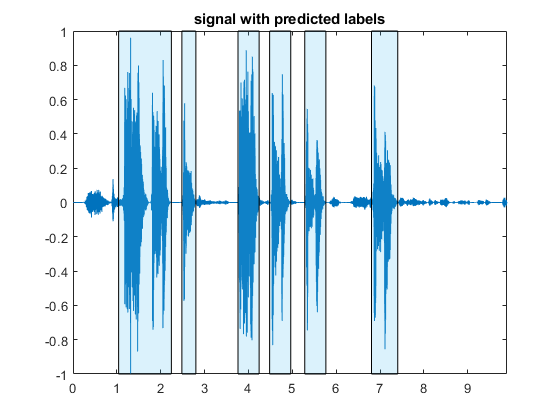
232



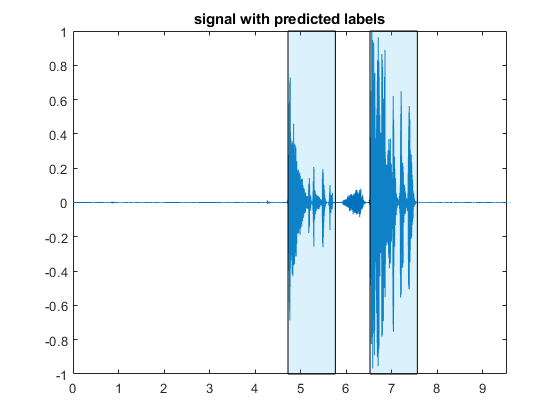
233



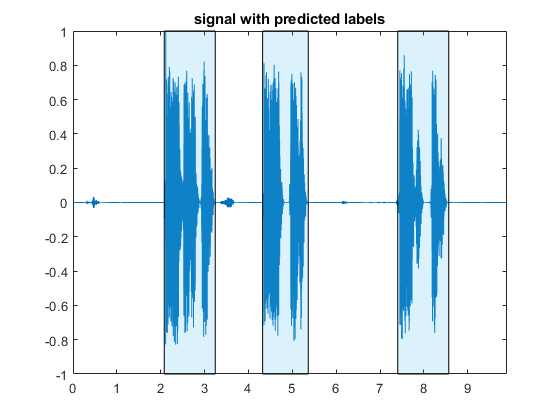
234



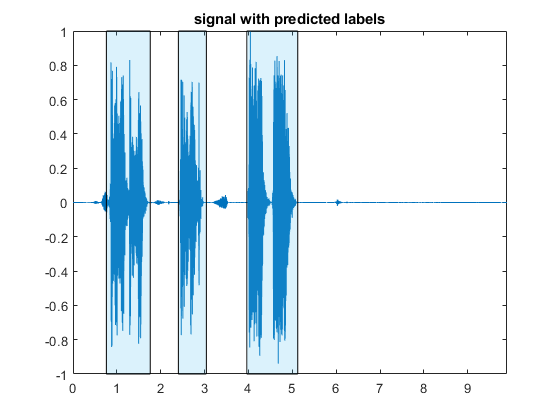
235



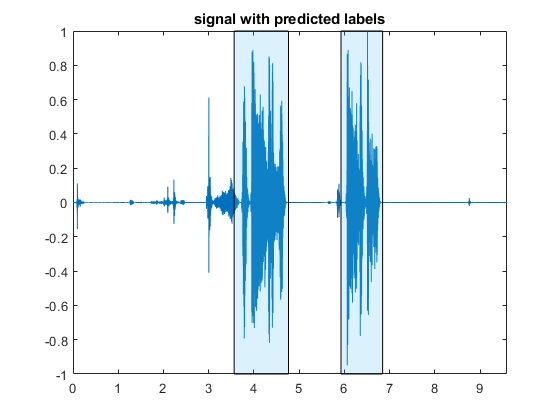
236



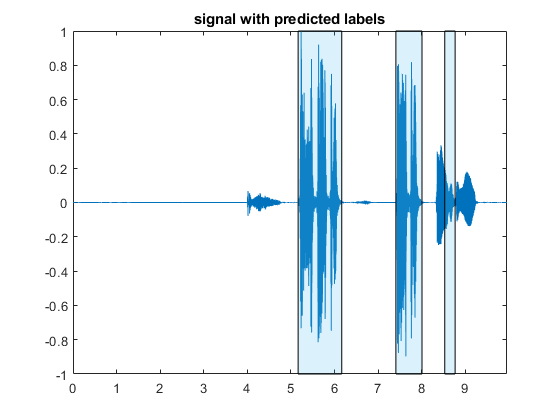
237



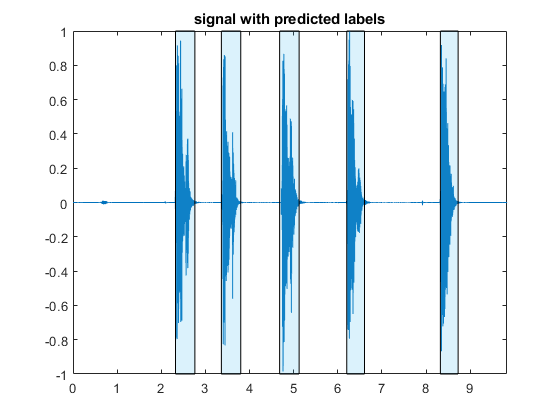
238



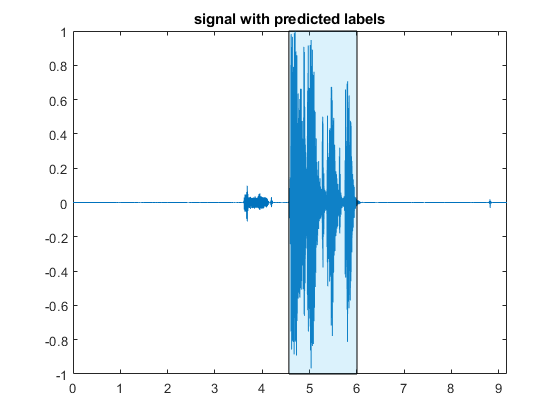
239



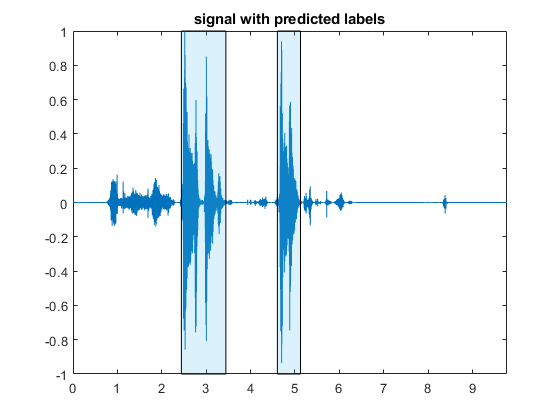
240



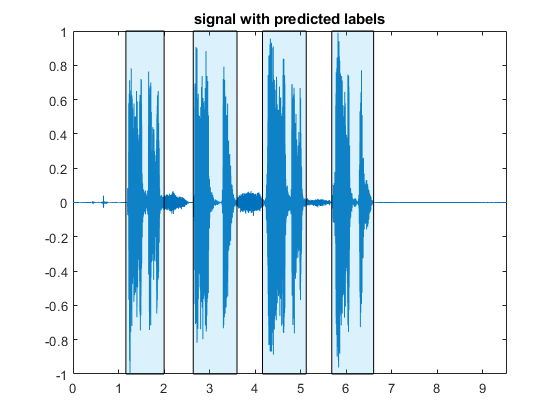
241



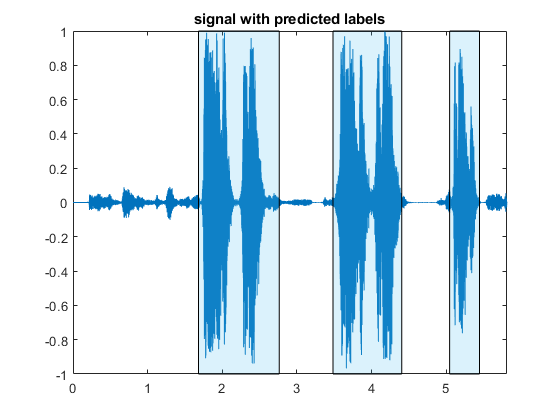
242



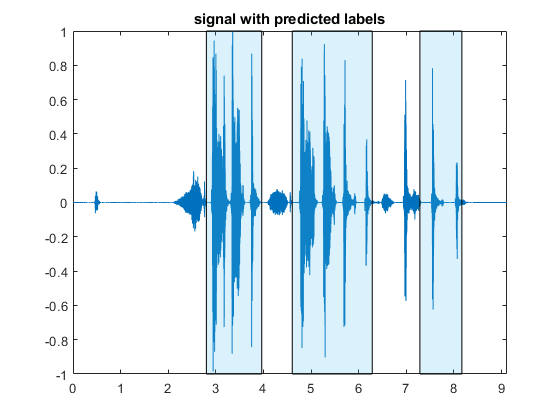
243



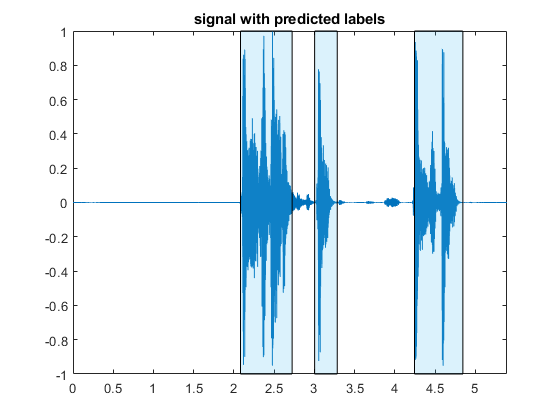
244



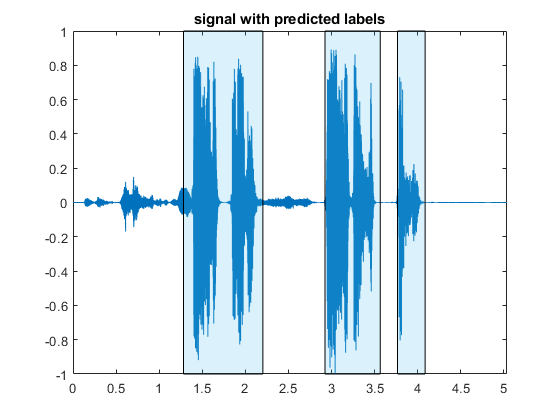
245



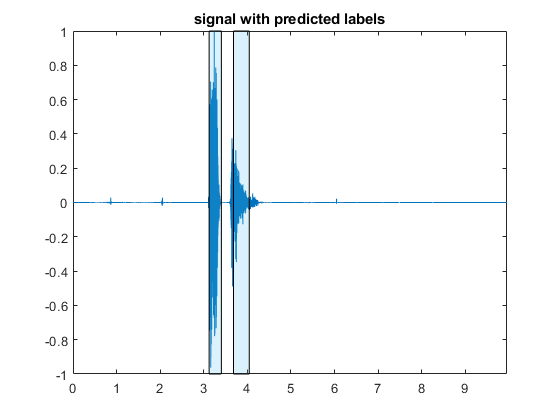
246



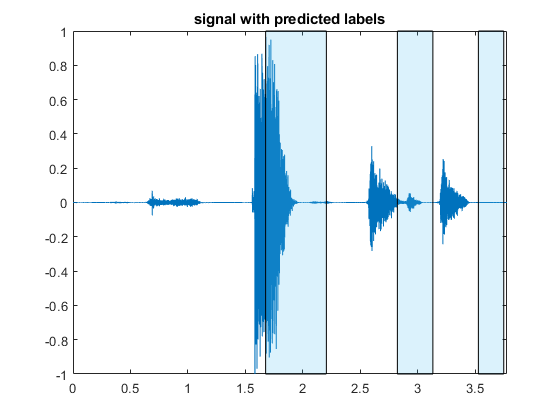
247



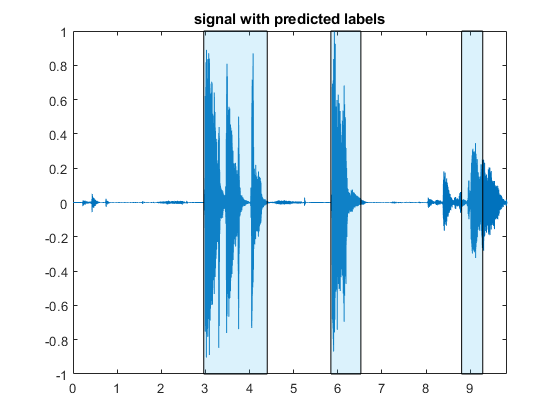
248



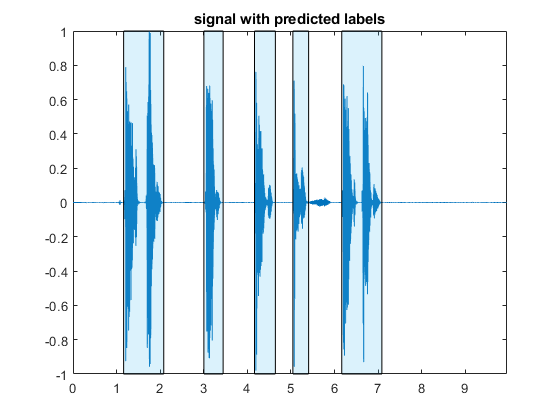
249



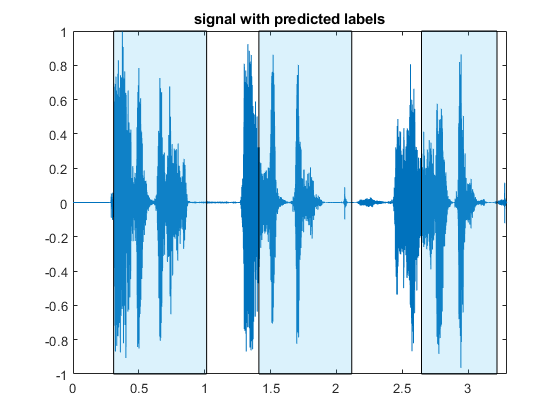
250



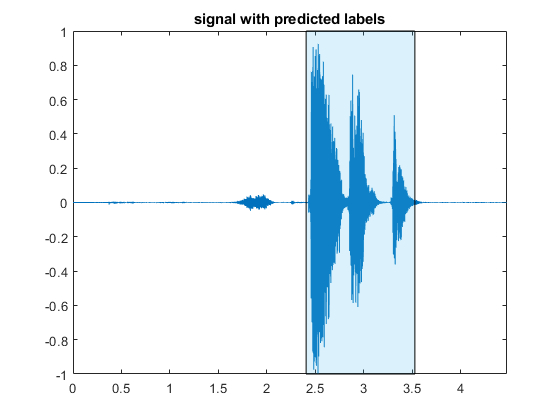
251



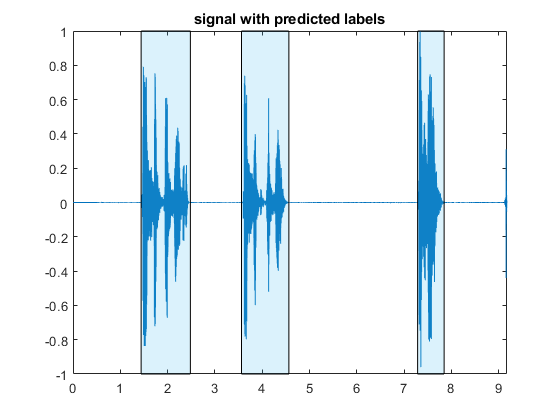
252



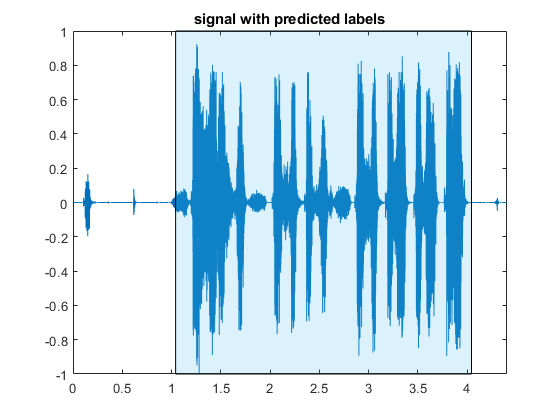
253



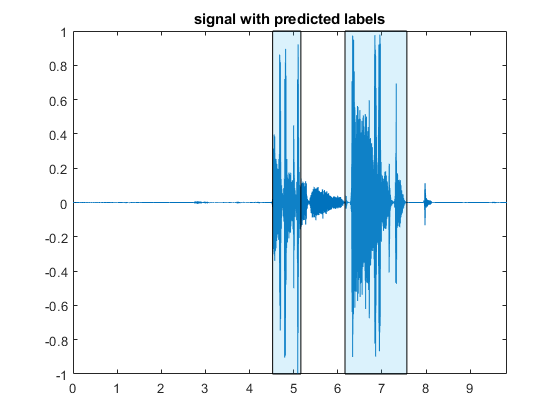
254



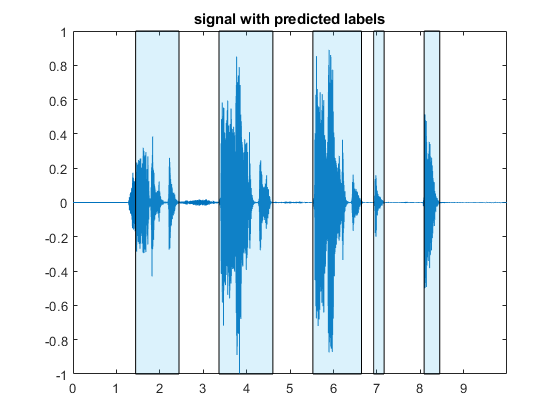
255



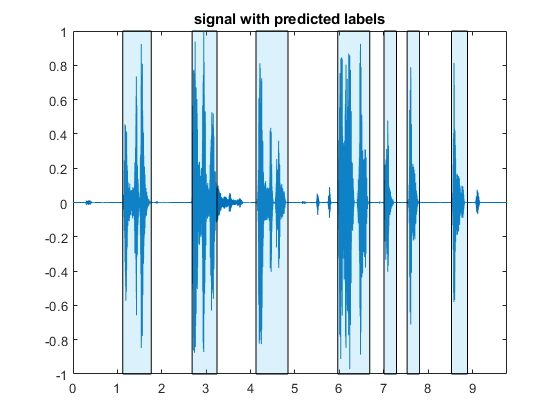
256



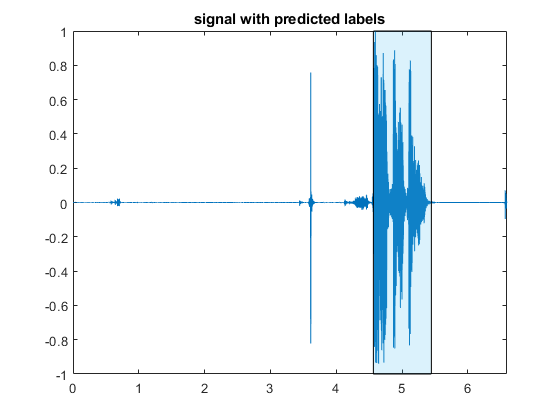
257



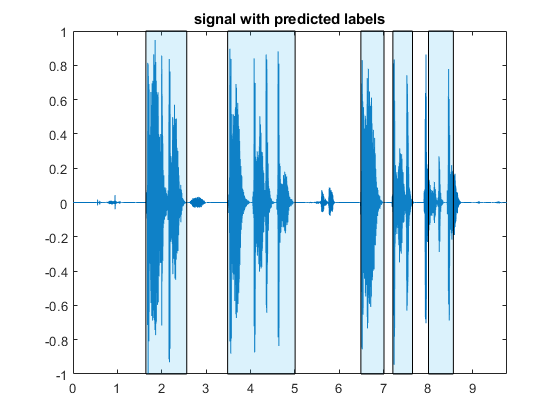
258



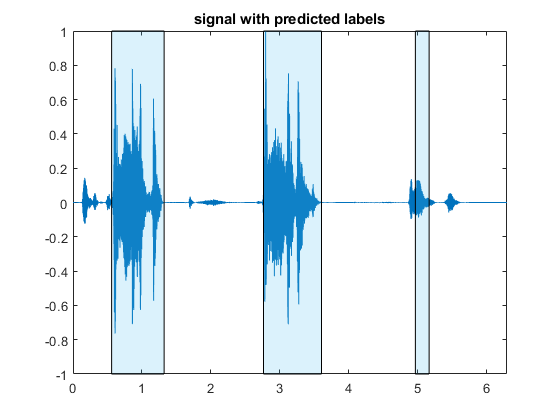
259



260



261



262