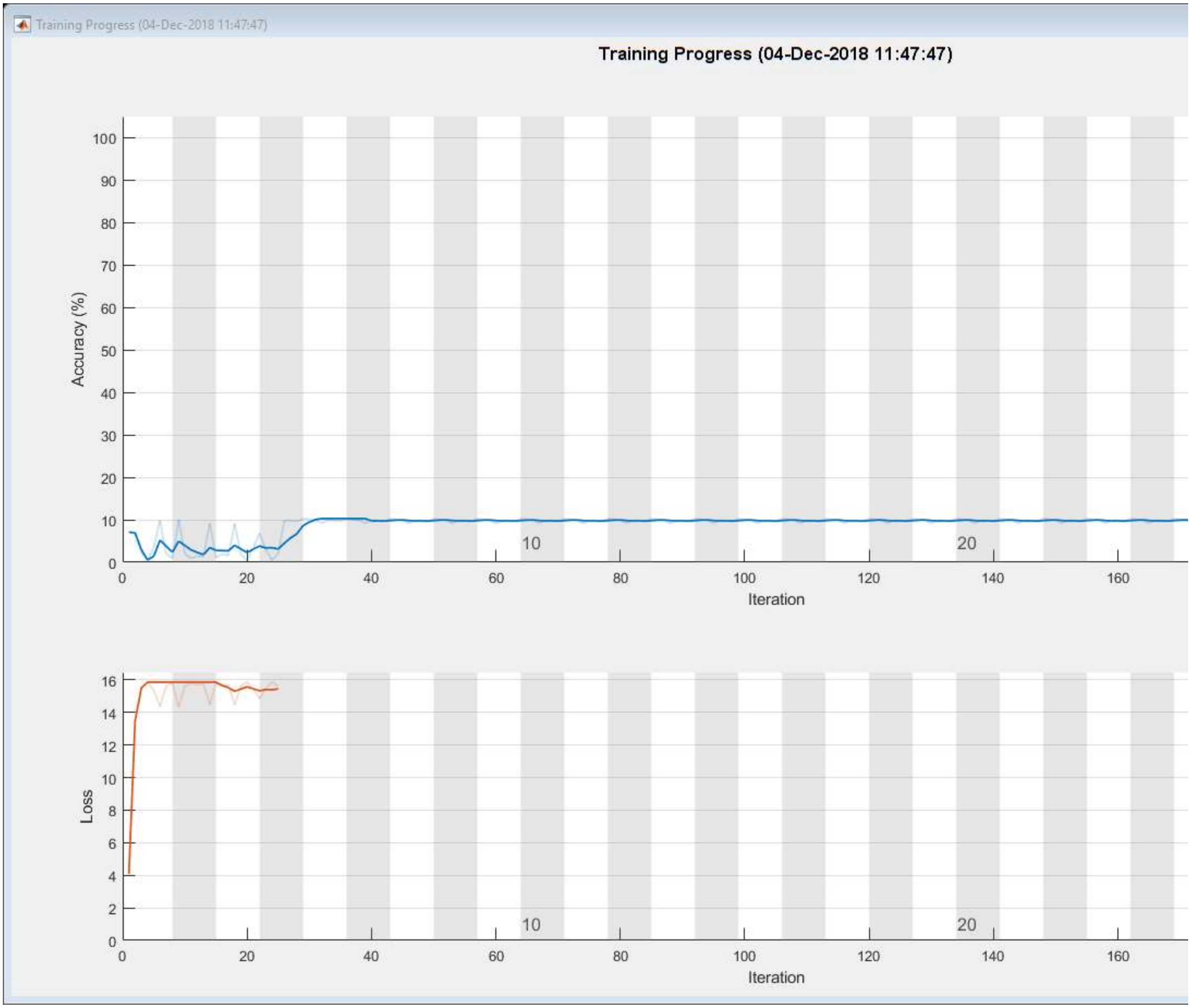
```
[imgDataTrain, labelsTrain, imgDataTest, labelsTest] = prepareData;
```

```
Preparing MNIST data...
MNIST data preparation complete.
```

```
warning off images:imshow:magnificationMustBeFitForDockedFigure
perm = randperm(numel(labelsTrain), 25);
subset = imgDataTrain(:,:,1,perm);
montage(subset)
```

```
layers = [  imageInputLayer([28 28 1])
            convolution2dLayer(5,20)
            reluLayer
            maxPooling2dLayer(2, 'Stride', 2)
            fullyConnectedLayer(10)
            softmaxLayer
            classificationLayer()   ]
miniBatchSize = 8192;
options = trainingOptions( 'sgdm',...
    'MiniBatchSize', miniBatchSize,...
    'Plots', 'training-progress');

net = trainNetwork(imgDataTrain, labelsTrain, layers, options);

predLabelsTest = net.classify(imgDataTest);
testAccuracy = sum(predLabelsTest == labelsTest) / numel(labelsTest)
```

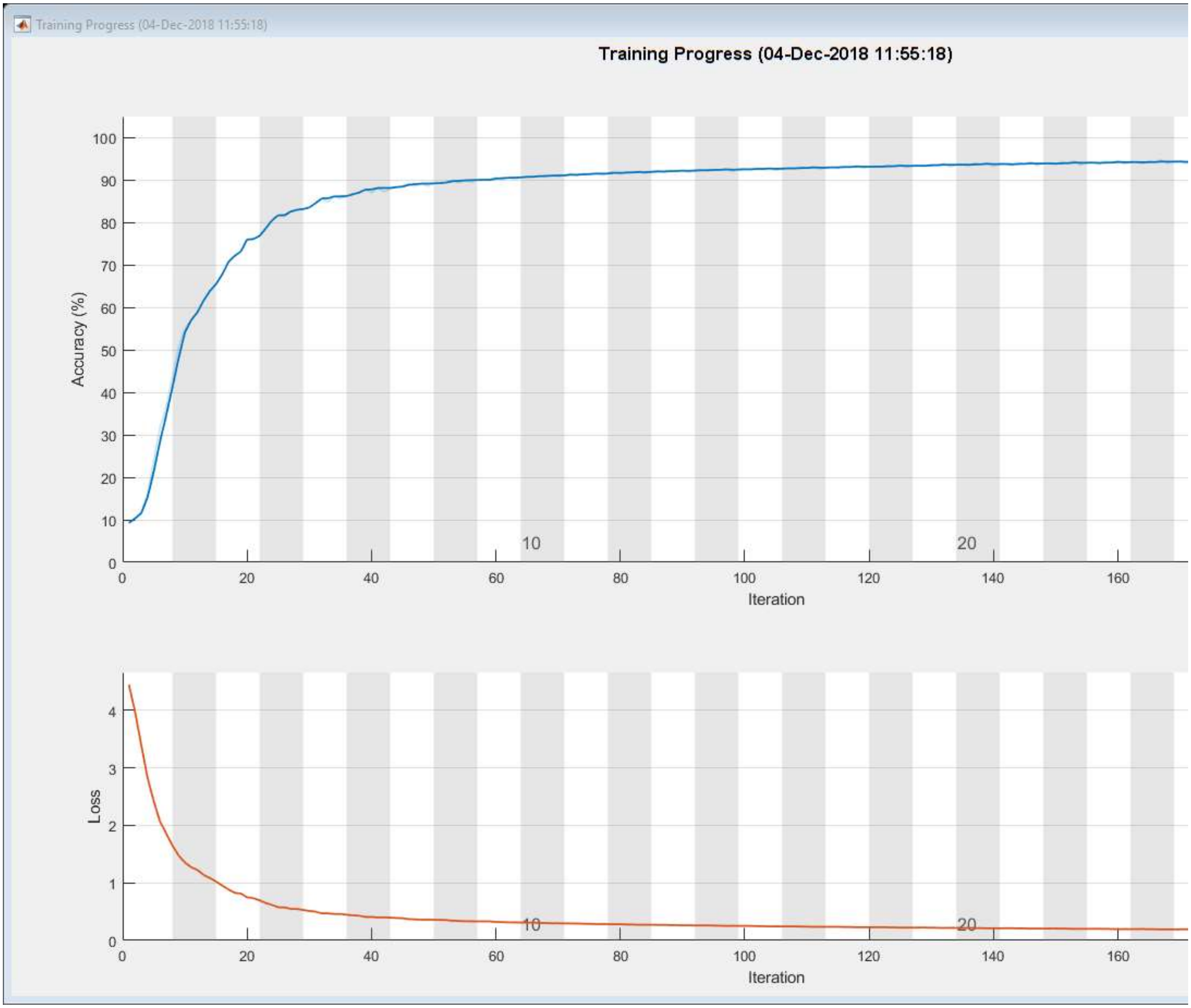
```
layers = 

  7x1 Layer array with layers:
     1   ''   Image Input             28x28x1 images with 'zerocenter' normalization
     2   ''   Convolution             20 5x5 convolutions with stride [1  1] and padding [0  0  0  0]
     3   ''   ReLU                    ReLU
     4   ''   Max Pooling             2x2 max pooling with stride [2  2] and padding [0  0  0  0]
     5   ''   Fully Connected         10 fully connected layer
     6   ''   Softmax                 softmax
     7   ''   Classification Output   crossentropyex
Training on single CPU.
Initializing image normalization.
|=========================================================================================|
|  Epoch  |  Iteration  |  Time Elapsed  |  Mini-batch  |  Mini-batch  |  Base Learning  |
|         |             |   (hh:mm:ss)   |   Accuracy   |     Loss     |      Rate       |
|=========================================================================================|
|       1 |           1 |       00:00:02 |        7.09% |       4.0873 |          0.0100 |
|       8 |          50 |       00:01:48 |       10.30% |          NaN |          0.0100 |
|      15 |         100 |       00:03:34 |       10.11% |          NaN |          0.0100 |
|      22 |         150 |       00:05:20 |        9.88% |          NaN |          0.0100 |
|      29 |         200 |       00:07:06 |        9.18% |          NaN |          0.0100 |
|      30 |         210 |       00:07:27 |        9.63% |          NaN |          0.0100 |
|=========================================================================================|

testAccuracy =

     0
```



```
options = trainingOptions( 'sgdm',...
    'MiniBatchSize', miniBatchSize,...
    'Plots', 'training-progress',...
    'InitialLearnRate', 0.0001);
net = trainNetwork(imgDataTrain, labelsTrain, layers, options);
```

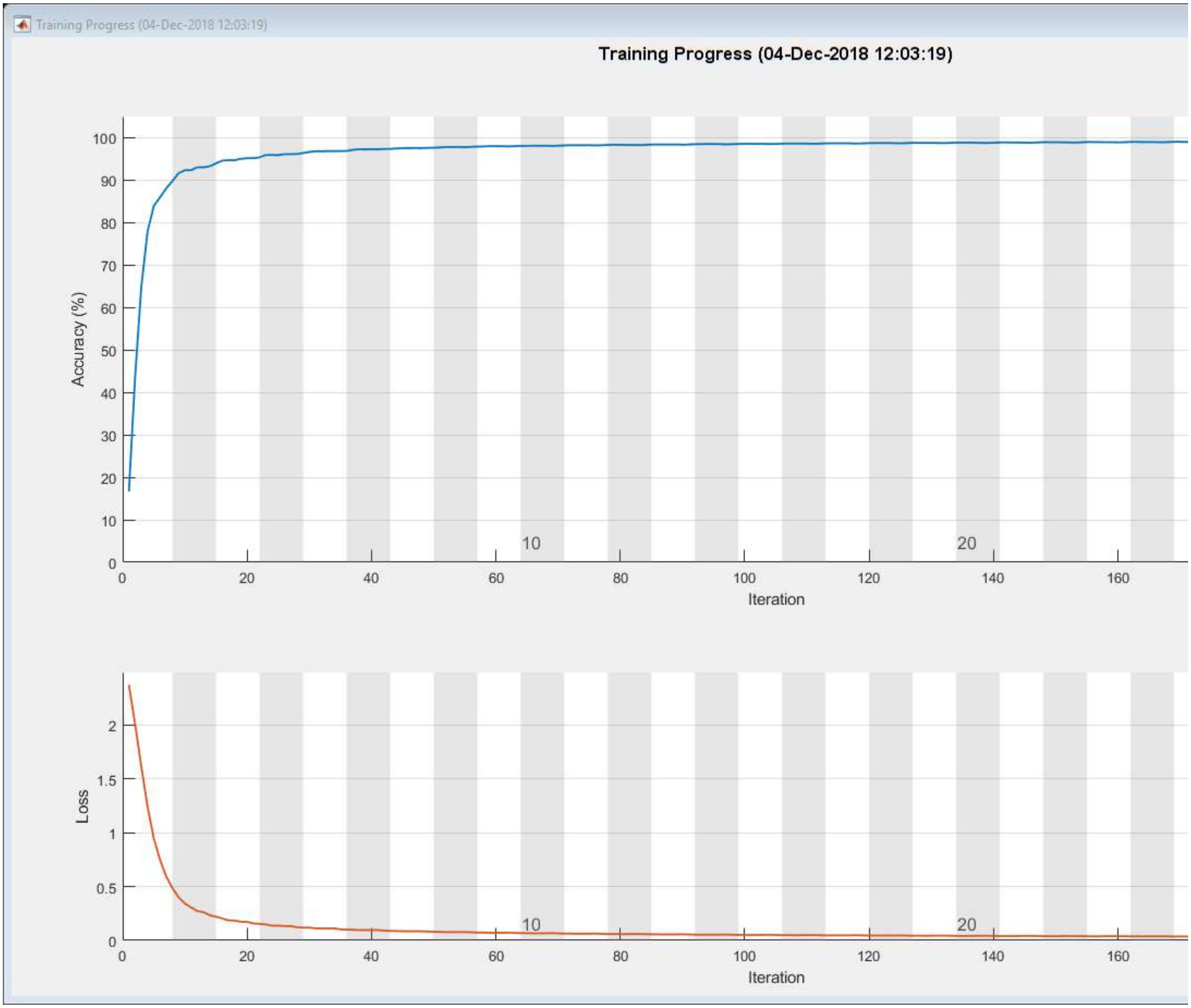
```
predLabelsTest = net.classify(imgDataTest);
testAccuracy = sum(predLabelsTest == labelsTest) / numel(labelsTest)
```

```
Training on single CPU.
Initializing image normalization.
|========================================================================================|
|  Epoch  |  Iteration  |  Time Elapsed  |  Mini-batch  |  Mini-batch  |  Base Learning  |
|         |             |   (hh:mm:ss)   |   Accuracy   |     Loss     |      Rate       |
|========================================================================================|
|       1 |           1 |       00:00:02 |        9.28% |       4.4432 |      1.0000e-04 |
|       8 |          50 |       00:01:54 |       89.27% |       0.3591 |      1.0000e-04 |
|      15 |         100 |       00:03:49 |       92.65% |       0.2571 |      1.0000e-04 |
|      22 |         150 |       00:05:42 |       93.71% |       0.2079 |      1.0000e-04 |
|      29 |         200 |       00:07:35 |       95.07% |       0.1727 |      1.0000e-04 |
|      30 |         210 |       00:07:58 |       94.54% |       0.1777 |      1.0000e-04 |
|========================================================================================|

testAccuracy =

    0.9523
```



```
layers = [
    imageInputLayer([28 28 1])
    convolution2dLayer(3,16,'Padding',1)
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(2,'Stride',2)
    convolution2dLayer(3,32,'Padding',1)
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(2,'Stride',2)
    convolution2dLayer(3,64,'Padding',1)
    batchNormalizationLayer
    reluLayer
    fullyConnectedLayer(10)
    softmaxLayer
    classificationLayer];
options = trainingOptions( 'sgdm',...
    'MiniBatchSize', miniBatchSize,...
    'Plots', 'training-progress');
net = trainNetwork(imgDataTrain, labelsTrain, layers, options);

predLabelsTest = net.classify(imgDataTest);
testAccuracy = sum(predLabelsTest == labelsTest) / numel(labelsTest)
```

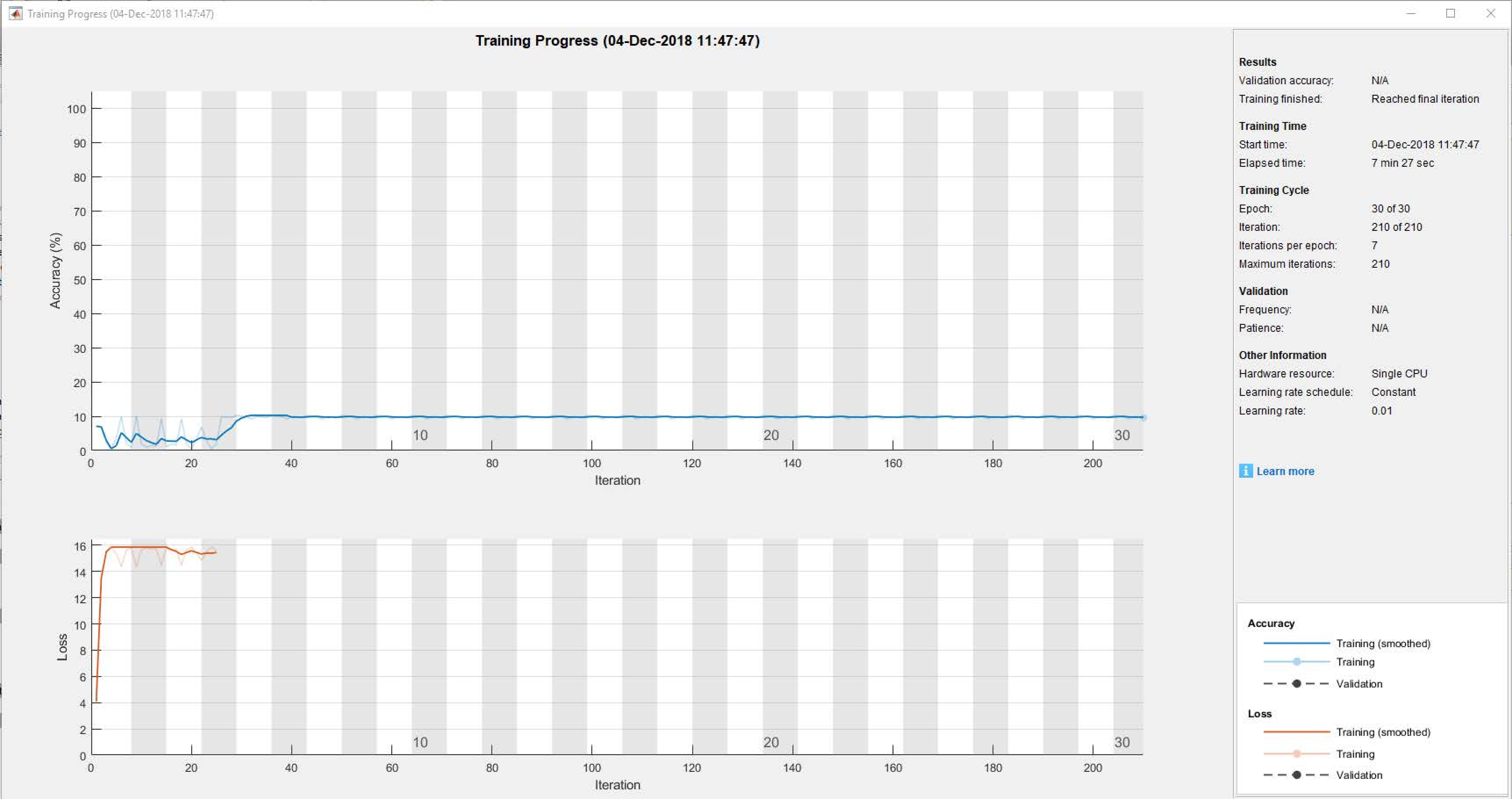
```
Training on single CPU.
Initializing image normalization.
|========================================================================================|
|  Epoch  |  Iteration  |  Time Elapsed  |  Mini-batch  |  Mini-batch  |  Base Learning  |
|         |             |   (hh:mm:ss)   |   Accuracy   |     Loss     |      Rate       |
|========================================================================================|
|       1 |           1 |       00:00:05 |       16.67% |       2.3717 |          0.0100 |
|       8 |          50 |       00:04:22 |       97.68% |       0.0760 |          0.0100 |
|      15 |         100 |       00:08:43 |       98.62% |       0.0545 |          0.0100 |
|      22 |         150 |       00:13:04 |       98.83% |       0.0389 |          0.0100 |
|      29 |         200 |       00:17:25 |       99.11% |       0.0364 |          0.0100 |
|      30 |         210 |       00:18:17 |       99.29% |       0.0301 |          0.0100 |
|========================================================================================|

testAccuracy =

    0.9888
```



```
[x,y]=meshgrid(unique(labelsTest),unique(labelsTest));
Pred=repmat(reshape(predLabelsTest,1,1,[]),numel(unique(labelsTest)),numel(unique(labelsTest)));
Actual=repmat(reshape(labelsTest,1,1,[]),numel(unique(labelsTest)),numel(unique(labelsTest)));
Confusion_Matrix=sum((((Actual==y)+(Pred==x))==2),3)
```

```
Confusion_Matrix =

  Columns 1 through 6

         973           0           1           0           0           0
           0        1133           1           1           0           0
           1           3        1021           1           2           0
           0           0           3         999           0           3
           0           0           0           0         976           0
           2           0           0           3           0         885
           8           2           2           0           1           2
           0           4           8           1           0           0
           3           0           2           0           1           1
           2           1           0           2           6           4

  Columns 7 through 10

           3           1           2           0
           0           0           0           0
           0           4           0           0
           0           3           1           1
           0           0           2           4
           1           1           0           0
         940           0           3           0
           0        1011           2           2
           1           0         963           3
           0           4           3         987
```

Published with MATLAB® R2018a



Training Progress (04-Dec-2018 11:47:47)

Training Progress (04-Dec-2018 11:47:47)

Results

Validation accuracy: N/A

Training finished: Reached final iteration

Training Time

Start time: 04-Dec-2018 11:47:47

Elapsed time: 7 min 27 sec

Training Cycle

Epoch: 30 of 30

Iteration: 210 of 210

Iterations per epoch: 7

Maximum iterations: 210

Validation

Frequency: N/A

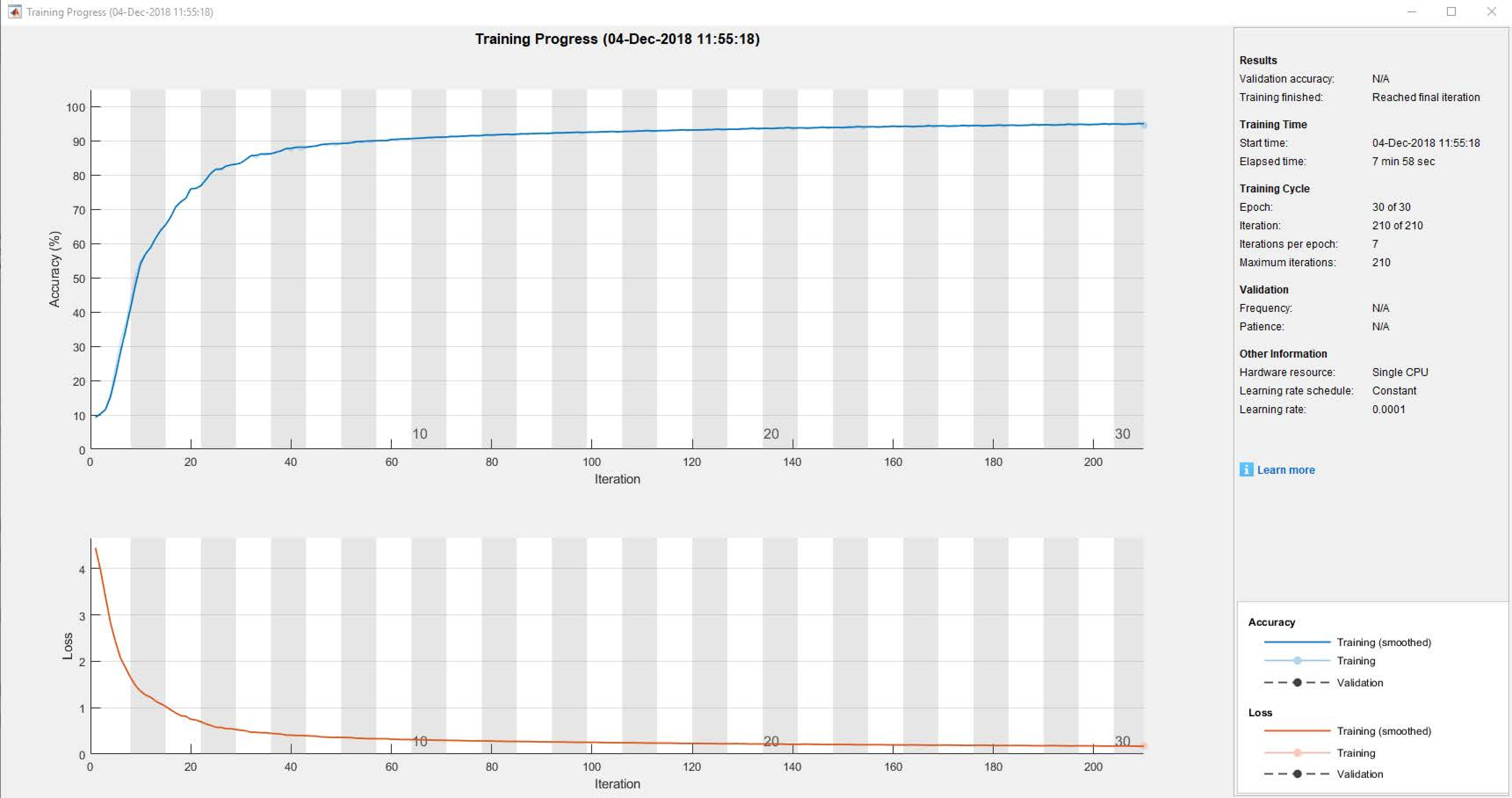
Patience: N/A

Other Information

Hardware resource: Single CPU

Learning rate schedule: Constant

Learning rate: 0.01



Training Progress (04-Dec-2018 11:55:18)

Results

Validation accuracy: N/A

Training finished: Reached final iteration

Training Time

Start time: 04-Dec-2018 11:55:18

Elapsed time: 7 min 58 sec

Training Cycle

Epoch: 30 of 30

Iteration: 210 of 210

Iterations per epoch: 7

Maximum iterations: 210

Validation

Frequency: N/A

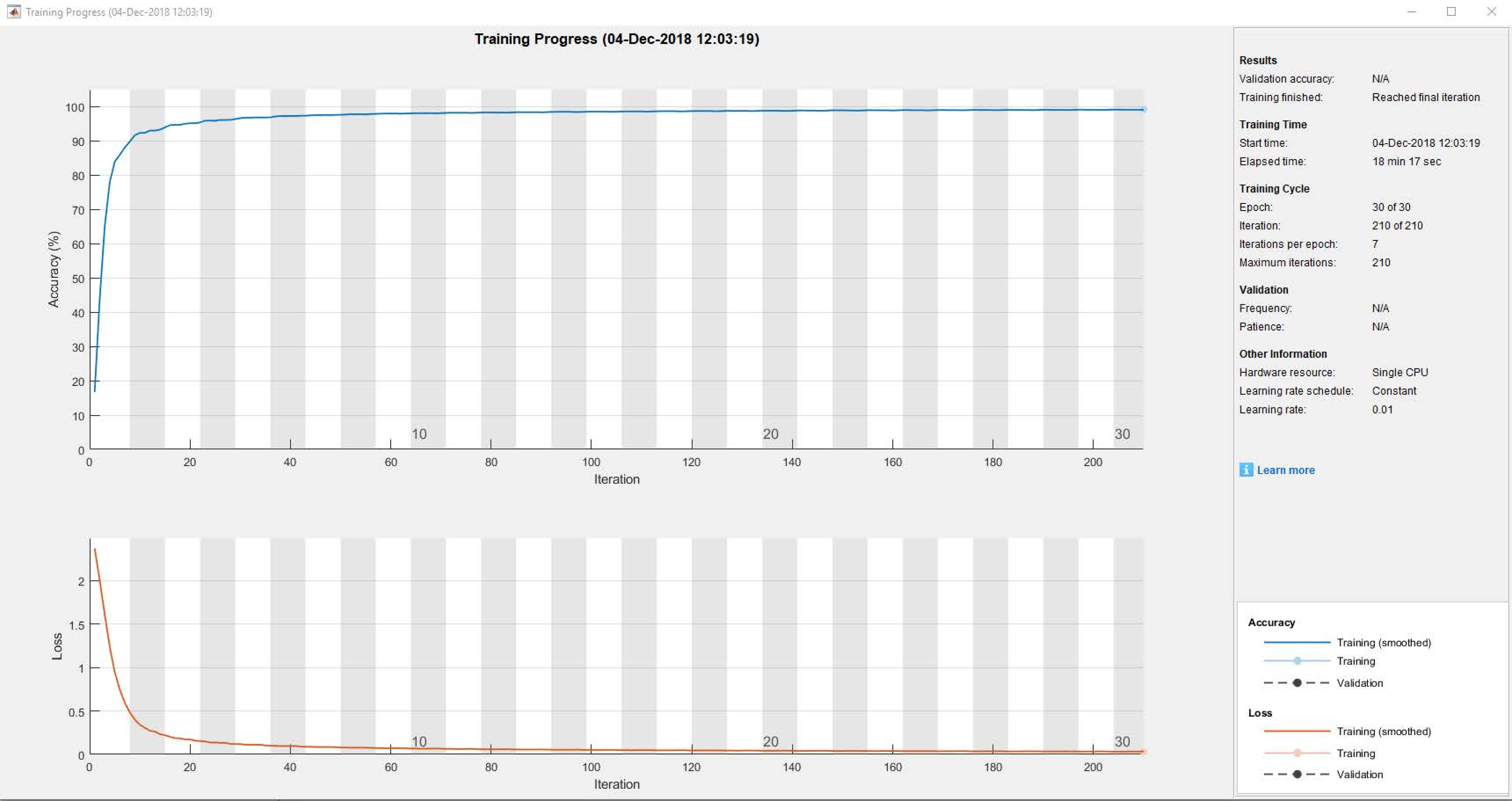
Patience: N/A

Other Information

Hardware resource: Single CPU

Learning rate schedule: Constant

Learning rate: 0.0001



Training Progress (04-Dec-2018 12:03:19)

Training Progress (04-Dec-2018 12:03:19)

Results

Validation accuracy: N/A

Training finished: Reached final iteration

Training Time

Start time: 04-Dec-2018 12:03:19

Elapsed time: 18 min 17 sec

Training Cycle

Epoch: 30 of 30

Iteration: 210 of 210

Iterations per epoch: 7

Maximum iterations: 210

Validation

Frequency: N/A

Patience: N/A

Other Information

Hardware resource: Single CPU

Learning rate schedule: Constant

Learning rate: 0.01