Artificial Intelligence

Transfer Learning

Garth Mortensen

2018.10.18

Pre-Processing

Prep workspace

```
clear all;
close all;
clc;
```

Data Loading

Read datastore

```
imds = imageDatastore(fullfile('C:\tmp\Iris_Imgs\'), ...
'IncludeSubfolders', true, 'FileExtensions', '.jpg', 'LabelSource', 'foldernames');
```

Shuffle and split into training and validation sets

```
[imdsTrain, imdsValidation] = splitEachLabel(imds, 0.7, 'randomized');
```

Modeling

Import pretrained **AlexNet** neural network

```
net = alexnet;
```

Summary of network layers

```
analyzeNetwork(net)
```

Set image size (rgb = 3)

```
imageSize = [227 227 3];
```

Extract all but last three layers from net (alexnet)

```
layersTransfer = net.Layers(1:end-6);
```

Replace the last three layers with:

1. Fully connected layer
2. Softmax layer
3. Classification output layer

```
numClasses = numel(categories(imdsTrain.Labels));
layers = [
    layersTransfer
    batchNormalizationLayer
    reluLayer
    fullyConnectedLayer(3)
    softmaxLayer
    classificationLayer];
```

Resize the images

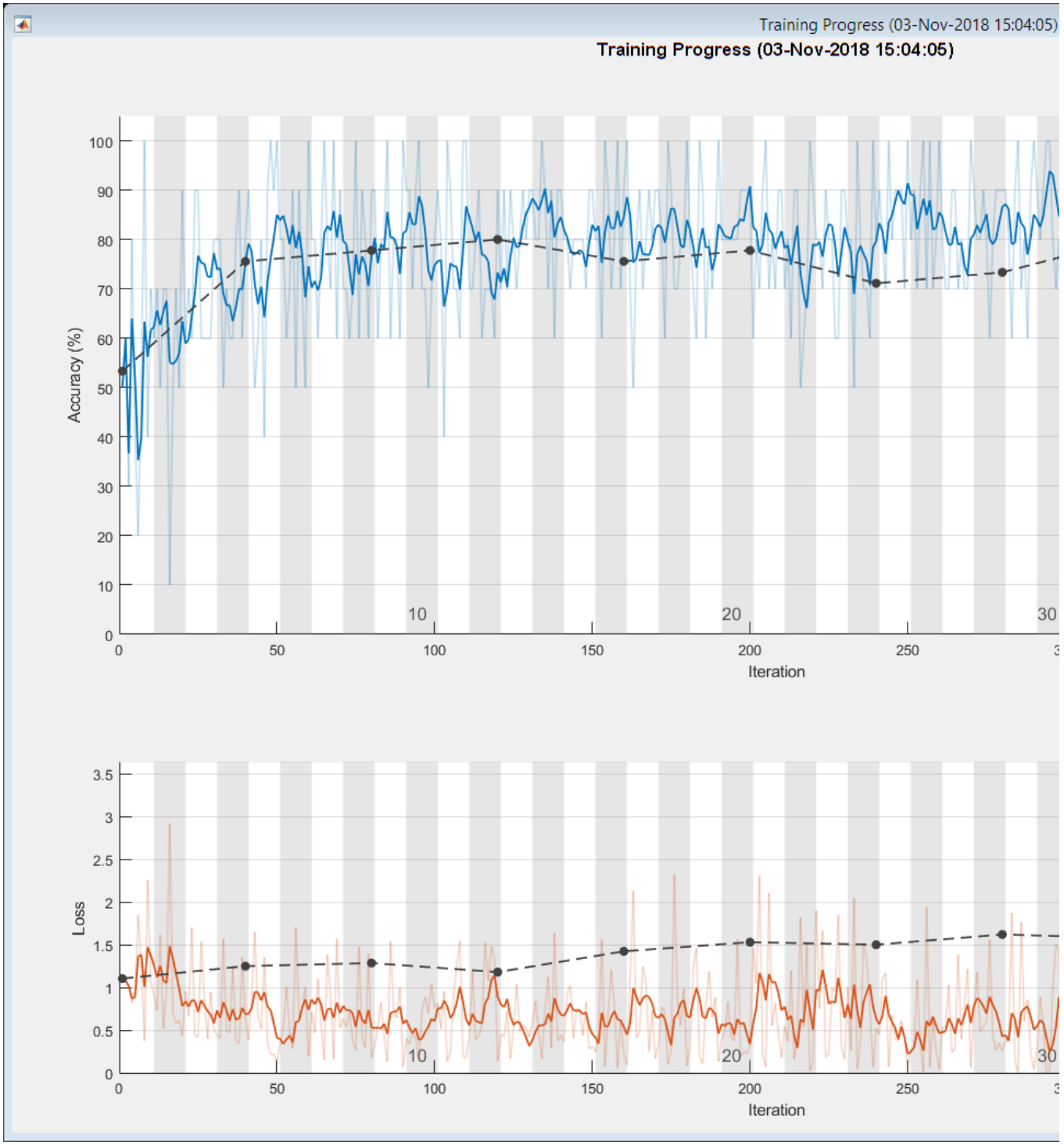
```
augimdsTrain       = augmentedImageDatastore(imageSize(1:2), imdsTrain);
augimdsValidation  = augmentedImageDatastore(imageSize(1:2), imdsValidation);
```

Specify training hyperparameters

```
options = trainingOptions('sgdm', ...
    'MaxEpochs', 40, ...
    'ValidationData', augimdsValidation, ...
    'ValidationFrequency', 40, ...
    'Verbose', false, ...
    'MiniBatchSize', 10,...
    'Plots', 'training-progress');
```

Train new network

```
netTransfer     = trainNetwork(augimdsTrain, layers, options);
```

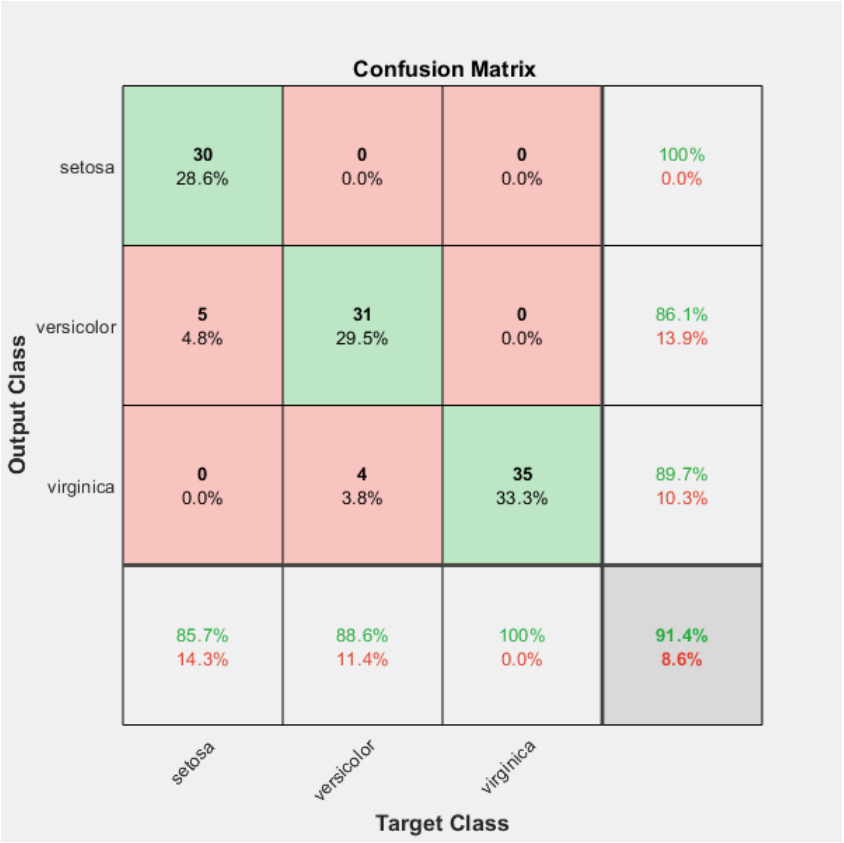


Obtain accuracy

```
YPred           = classify(netTransfer, augimdsTrain);
YValidation     = imdsTrain.Labels;
imdsAccuracy    = sum(YPred == YValidation) / numel(YValidation);
```

Confusion matrix

```
plotconfusion(YValidation, YPred)
```



Save the workspace

```
save('C:\tmp\netTransfer.mat', 'netTransfer', 'YPred', 'YValidation')

%clear all
%load('C:\tmp\netTransfer.mat')
```