**一、使用10行MATLAB代码尝试深度学习**

1.运行以下命令，并获得预训练的神经网络。

net = alexnet; % Load the neural network

需要先使用附加功能资源管理器下载Deep Learning Toolbox Model for AlexNet Network。AlexNet 是预训练的卷积神经网络 (CNN)，已基于超过一百万个图像进行训练，可以将图像分为 1000 个对象类别（例如键盘、鼠标、咖啡杯、铅笔和多种动物）。

2.用手机拍摄鼠标、杯子等照片，并导入到Matlab中。

3. 执行以下命令，对图片中的物体进行识别

im=imread('ss.png'); % Take a picture

image(im); % Show the picture

im = imresize(im,[227 227]); % Resize the picture for alexnet

label = classify(net,im); % Classify the picture

title(char(label)); % Show the class label

drawnow

**二、对比GoogleNet网络对图片的分类**

GoogLeNet 已经对超过一百万个图像进行了训练，可以将图像分为 1000 个对象类别（例如键盘、咖啡杯、铅笔和多种动物）。该网络已基于大量图像学习了丰富的特征表示。网络以图像作为输入，然后输出图像中对象的标签以及每个对象类别的概率。

net = googlenet;

I = imread('ss.png');

inputSize = net.Layers(1).InputSize;

I = imresize(I,inputSize(1:2));

label = classify(net,I);

figure

imshow(I)

title(string(label))

**三、使用深度网络设计器创建简单的序列分类网络**

**1．加载数据**

按照元音数据集。预测变量是包含不同长度序列的元胞数组，**特征维度为12**。标签是由**标签 1、2、...、9** 组成的分类向量。

[XTrain,YTrain] = japaneseVowelsTrainData;

[XValidation,YValidation] = japaneseVowelsTestData;

**2.查看前几个训练序列的大小。**

序列是具有 12 行（每个特征一行）和不同列数（每个时间步一列）的矩阵。

XTrain(1:5)

**3.进行网络设计**

deepNetworkDesigner （也可从APP中打开）

或者用

layers = [

imageInputLayer(inputSize)

convolution2dLayer(5,20)

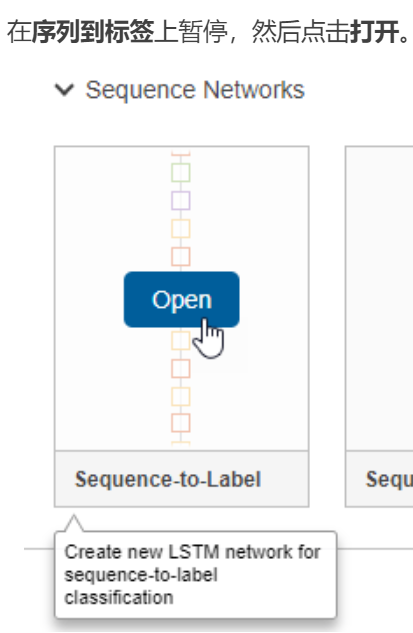
batchNormalizationLayer

reluLayer

fullyConnectedLayer(numClasses)

softmaxLayer

classificationLayer];

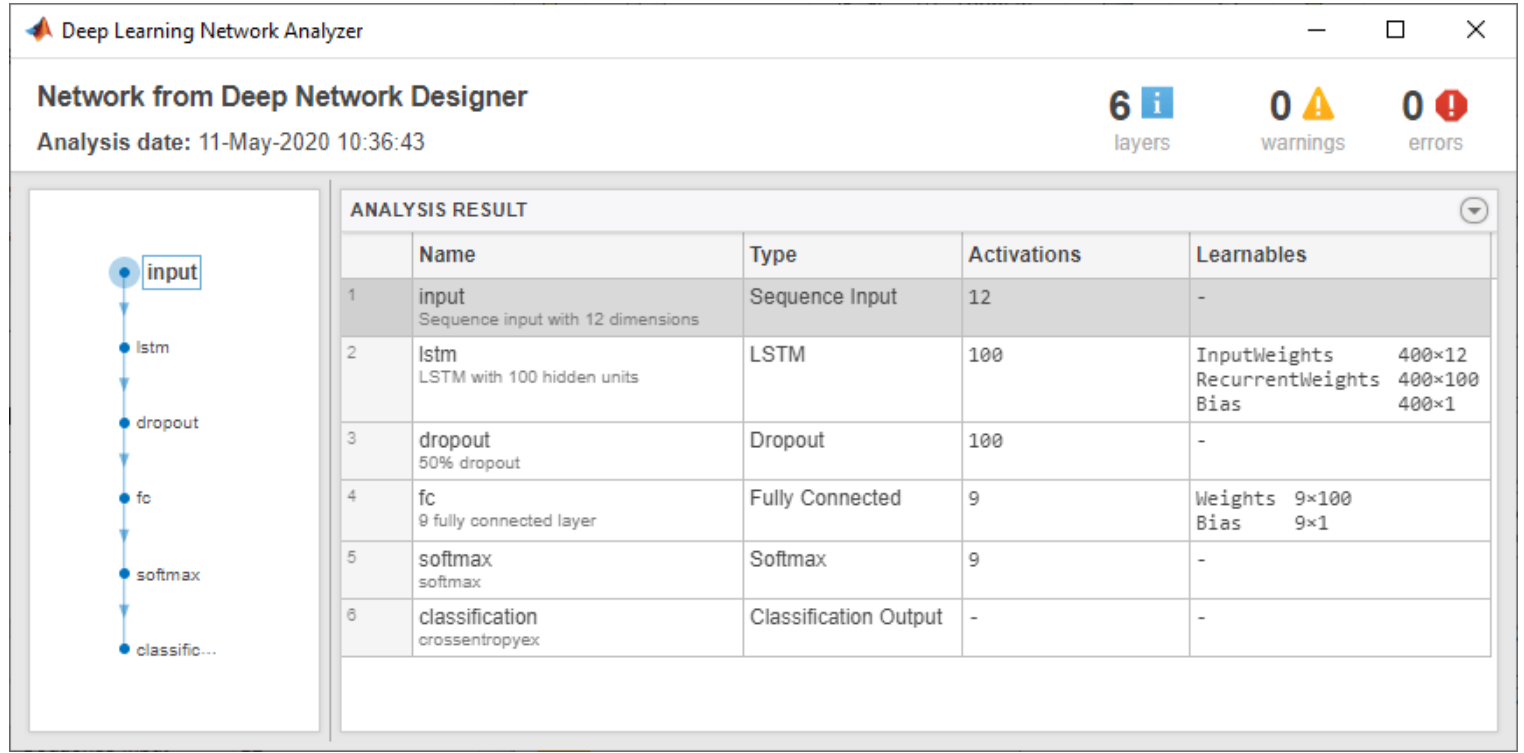


选择 **sequenceInputLayer**，检查并确认 **InputSize** 设置为 12，与特征维度匹配。

选择 fullyConnectedLayer，检查并确认 OutputSize 设置为 9，即类的数目。

**4.检查网络架构**

要检查网络并查看层的详细信息，请点击分析。



5.**导出网络架构**

要将网络架构导出到工作区，请在设计器选项卡上，点击导出。深度网络设计器将网络保存为变量 layers\_1。

也可以通过选择导出 > 生成代码来生成用于构造网络架构的代码。

**6.训练网络**

指定训练选项并训练网络。

由于小批量数据存储较小且序列较短，因此更适合在 CPU 上训练。将'ExecutionEnvironment' 设置为 'cpu'。要在 GPU（如果可用）上进行训练，请将 'ExecutionEnvironment' 设置为 'auto'（默认值）。

miniBatchSize = 27;

options = trainingOptions('adam', ...

'ExecutionEnvironment','cpu', ...

'MaxEpochs',100, ...

'MiniBatchSize',miniBatchSize, ...

'ValidationData',{XValidation,YValidation}, ...

'GradientThreshold',2, ...

'Shuffle','every-epoch', ...

'Verbose',false, ...

'Plots','training-progress');

net = trainNetwork(XTrain,YTrain,layers\_1,options);

YPred = classify(net,XValidation,'MiniBatchSize',miniBatchSize);

acc = mean(YPred == YValidation)

**四、针对三中跟换为bilstm网络或者更改隐藏层的节点数，看看分类效果**

注意bilstm中的输出形式改为last

net = trainNetwork(XTrain,YTrain,layers\_2,options);

YPred = classify(net,XValidation,'MiniBatchSize',miniBatchSize);

acc = mean(YPred == YValidation)

**五、超深超分辨率 (VDSR) 神经网络从低分辨率图像估计高分辨率图像**

**1.载入超分辨率模型**

load('trainedVDSRNet.mat');

**2. 创建一个低分辨率图像**

用于比较使用深度学习的超分辨率结果和使用双三次插值等传统图像处理方法的结果。测试数据集 testImages 包含 Image Processing Toolbox中提供的21个未失真图像。将这些图像加载到 imageDatastore 中。

exts = {'.jpg','.png'};

fileNames = {'sherlock.jpg','car2.jpg','fabric.png','greens.jpg','hands1.jpg','kobi.png', ...

'lighthouse.png','micromarket.jpg','office\_4.jpg','onion.png','pears.png','yellowlily.jpg', ...

'indiancorn.jpg','flamingos.jpg','sevilla.jpg','llama.jpg','parkavenue.jpg', ...

'peacock.jpg','car1.jpg','strawberries.jpg','wagon.jpg'};

filePath = [fullfile(matlabroot,'toolbox','images','imdata') filesep];

filePathNames = strcat(filePath,fileNames);

testImages = imageDatastore(filePathNames,'FileExtensions',exts);

**3. 测试图像显示**

montage(testImages)

**4. 选择其中一个图像作为超分辨率的参考图像**

indx = 2; % Index of image to read from the test image datastore

Ireference = readimage(testImages,indx);

Ireference = im2double(Ireference);

figure

imshow(Ireference)

title('High-Resolution Reference Image')

**5. 创建参考图像的低分辨率版本**

使用 [imresize](https://localhost:31515/static/help/images/ref/imresize.html) (Image Processing Toolbox) 和缩放因子 0.25 创建高分辨率参考图像的一个低分辨率版本。图像的高频分量在缩减分辨率过程中丢失。

scaleFactor = 0.25;

Ilowres = imresize(Ireference,scaleFactor,'bicubic');

figure

imshow(Ilowres)

title('Low-Resolution Image')

**6. 使用双三次插值提高图像分辨率**

[nrows,ncols,np] = size(Ireference);

Ibicubic = imresize(Ilowres,[nrows ncols],'bicubic');

figure

imshow(Ibicubic)

title('High-Resolution Image Obtained Using Bicubic Interpolation')

**7. 使用预训练的 VDSR 网络提高图像分辨率**

使用 [rgb2ycbcr](https://localhost:31515/static/help/images/ref/rgb2ycbcr.html) (Image Processing Toolbox) 函数将低分辨率图像从 RGB 颜色空间转换为亮度 (Iy) 和色度（Icb 和 Icr）通道。

Iycbcr = rgb2ycbcr(Ilowres);

Iy = Iycbcr(:,:,1);

Icb = Iycbcr(:,:,2);

Icr = Iycbcr(:,:,3);

使用双三次插值扩增亮度通道和两个色度通道。上采样的色度通道 Icb\_bicubic 和 Icr\_bicubic 不需要进一步处理。

Iy\_bicubic = imresize(Iy,[nrows ncols],'bicubic');

Icb\_bicubic = imresize(Icb,[nrows ncols],'bicubic');

Icr\_bicubic = imresize(Icr,[nrows ncols],'bicubic');

对扩增的亮度分量 Iy\_bicubic 应用经过训练的 VDSR 网络。观察来自最终层（回归层）的[activations](https://localhost:31515/static/help/deeplearning/ref/seriesnetwork.activations.html)。网络的输出是所需的残差图像。

Iresidual = activations(net,Iy\_bicubic,41);

Iresidual = double(Iresidual);

figure

imshow(Iresidual,[])

title('Residual Image from VDSR')

将残差图像与扩增的亮度分量相加，得到高分辨率 VDSR 亮度分量。

Isr = Iy\_bicubic + Iresidual;

将高分辨率 VDSR 亮度分量与扩增的颜色分量串联起来。使用 [ycbcr2rgb](https://localhost:31515/static/help/images/ref/ycbcr2rgb.html) (Image Processing Toolbox) 函数将图像转换为 RGB 颜色空间。结果为使用 VDSR 得到的最终高分辨率彩色图像。

Ivdsr = ycbcr2rgb(cat(3,Isr,Icb\_bicubic,Icr\_bicubic));

figure

imshow(Ivdsr)

title('High-Resolution Image Obtained Using VDSR')

8. 图像评价

每个图像的峰值信噪比 (PSNR)，PSNR 值越大，通常表示图像质量越好。

bicubicPSNR = psnr(Ibicubic,Ireference)

vdsrPSNR = psnr(Ivdsr,Ireference)

每个图像的结构相似性指数 (SSIM)，SSIM 对照参考图像评估图像三个特性的视觉效果：亮度、对比度和结构。SSIM 值越接近 1，测试图像与参考图像越一致。

bicubicSSIM = ssim(Ibicubic,Ireference)

vdsrSSIM = ssim(Ivdsr,Ireference)

使用自然图像质量评价方法 (NIQE) 测量图像感知质量。NIQE 分数越小，表示感知质量越好。

bicubicNIQE = niqe(Ibicubic)

vdsrNIQE = niqe(Ivdsr)

**六、简单的甲状腺图像分割**

1.载入训练集和标签

imageFolderTrain='E:\matlab\_code\chuangxinke\imgs';

labelFolderTrain='E:\matlab\_code\chuangxinke\labels';

imdsTrain\_ZX = imageDatastore(imageFolderTrain);

classNames = ["thyroid" "background"];

labels = [1 0];

pxdsTrain\_ZX = pixelLabelDatastore(labelFolderTrain,classNames,labels);

2. 合并训练集和标签

ds = combine(imdsTrain\_ZX,pxdsTrain\_ZX);

tbl = countEachLabel(pxdsTrain\_ZX)

3. 计算甲状腺区域的像素，并分配训练权重

numberPixels = sum(tbl.PixelCount);

numberPixels = sum(tbl.PixelCount);

frequency = tbl.PixelCount / numberPixels;

classWeights = 1 ./ frequency;

4. 构建网络结构

inputSize = [224 224 3];

filterSize = 3;

numFilters = 32;

numClasses = numel(classNames);

layers = [

imageInputLayer(inputSize)

convolution2dLayer(filterSize,numFilters,'DilationFactor',1,'Padding','same')

batchNormalizationLayer

reluLayer

convolution2dLayer(filterSize,numFilters,'DilationFactor',2,'Padding','same')

batchNormalizationLayer

reluLayer

convolution2dLayer(filterSize,numFilters,'DilationFactor',4,'Padding','same')

batchNormalizationLayer

reluLayer

convolution2dLayer(1,numClasses)

softmaxLayer

pixelClassificationLayer('Classes',classNames,'ClassWeights',classWeights)];

4.设置训练选项以及超参数

options = trainingOptions('sgdm', ...

'MaxEpochs', 100, ...

'MiniBatchSize', 64, ...

'InitialLearnRate', 1e-3);

或者更为详细的

initialLearningRate = 0.05;

maxEpochs = 150;

minibatchSize = 32;

l2reg = 0.0001;

options = trainingOptions('sgdm',...

'InitialLearnRate',initialLearningRate, ...

'Momentum',0.9,...

'L2Regularization',l2reg,...

'MaxEpochs',maxEpochs,...

'MiniBatchSize',minibatchSize,...

'LearnRateSchedule','piecewise',...

'LearnRateDropFactor',0.5, ...

'LearnRateDropPeriod',20, ...

'Shuffle','every-epoch',...

'GradientThresholdMethod','l2norm',...

'GradientThreshold',0.05, ...

'Plots','training-progress', ...

'VerboseFrequency',20);

5. 训练网络

net = trainNetwork(ds,layers,options);

6. 测试其中的一个图像

imgTest = imread('imgs/201612161226530246SMP.bmp');

figure

imshow(imgTest)

C = semanticseg(imgTest,net);

B = labeloverlay(imgTest,C);

figure

imshow(B)

**七、变化特征的简单甲状腺图像分割**

1.载入训练集和标签

imageFolderTrain='E:\matlab\_code\chuangxinke\imgs';

labelFolderTrain='E:\matlab\_code\chuangxinke\labels';

imdsTrain\_ZX = imageDatastore(imageFolderTrain);

classNames = ["thyroid" "background"];

labels = [1 0];

pxdsTrain\_ZX = pixelLabelDatastore(labelFolderTrain,classNames,labels);

特征改进

imdsImp\_ZX = transform(imdsTrain\_ZX,@(x) imgprocessing(x));

2. 合并训练集和标签

ds = combine(imdsImp\_ZX,pxdsTrain\_ZX);

tbl = countEachLabel(pxdsTrain\_ZX)

3. 计算甲状腺区域的像素，并分配训练权重

numberPixels = sum(tbl.PixelCount);

frequency = tbl.PixelCount / numberPixels;

classWeights = 1 ./ frequency;

4. 构建网络结构

inputSize = [224 224 1];

filterSize = 3;

numFilters = 32;

numClasses = numel(classNames);

layers = [

imageInputLayer(inputSize)

convolution2dLayer(filterSize,numFilters,'DilationFactor',1,'Padding','same')

batchNormalizationLayer

reluLayer

convolution2dLayer(filterSize,numFilters,'DilationFactor',2,'Padding','same')

batchNormalizationLayer

reluLayer

convolution2dLayer(filterSize,numFilters,'DilationFactor',4,'Padding','same')

batchNormalizationLayer

reluLayer

convolution2dLayer(1,numClasses)

softmaxLayer

pixelClassificationLayer('Classes',classNames,'ClassWeights',classWeights)];

4.设置训练选项以及超参数

options = trainingOptions('sgdm', ...

'MaxEpochs', 100, ...

'MiniBatchSize', 64, ...

'InitialLearnRate', 1e-3);

或者更为详细的

initialLearningRate = 0.05;

maxEpochs = 150;

minibatchSize = 16;

l2reg = 0.0001;

options = trainingOptions('sgdm',...

'InitialLearnRate',initialLearningRate, ...

'Momentum',0.9,...

'L2Regularization',l2reg,...

'MaxEpochs',maxEpochs,...

'MiniBatchSize',minibatchSize,...

'LearnRateSchedule','piecewise',...

'LearnRateDropFactor',0.5, ...

'LearnRateDropPeriod',30, ...

'Shuffle','every-epoch',...

'GradientThresholdMethod','l2norm',...

'GradientThreshold',0.05, ...

'Plots','training-progress', ...

'VerboseFrequency',20);

5. 训练网络

net = trainNetwork(ds,layers,options);

6. 测试其中的一个图像

imgTest = imread('imgs/201612161226530246SMP.bmp');

figure

imshow(imgTest)

C = semanticseg(imgprocessing(imgTest),net);

B = labeloverlay(imgTest,C);

figure

imshow(B)

https://ww2.mathworks.cn/help/vision/ug/getting-started-with-semantic-segmentation-using-deep-learning.html