

程序说明

实例 4.2-1 功能：对输入图像中数字的倾斜角度进行预测，计算预测准确率和均方根误差(RMSE) 作者：zhaoxch_mail@sina.com 时间：2020年2月29日 版本：DLTEX2-V1 注：1) 本实例主要用于说明如何对卷积网络进行训练？改变训练参数后的影响。所以，请重点关注步骤3、步骤4 2) 要改变训练参数，主要在步骤3配置训练选项中修改 3) 读者可以结合注释（%之后的语句）对程序进行理解

编程实践1：改变配置训练选项中的参数，看训练效果如何？ 1) 将初始学习率改为0.01，看效果如何？（方法：将本例中69行的0.001改为0.01。） 2) 采用ADAM的训练方法，看效果如何？（方法：将本例中第66行sgdm改为adam。） 3) 去掉Dropout层，看效果如何？（方法：将本例中第59行 dropoutLayer(0.2)，之前加%，将其注释掉）编程实践2：在加上一个卷积层，看预测效果如何？（方法：将第55-57行之前的%去掉。）

清除内存、清除屏幕

```
clear
clc
```

步骤1：加载和显示图像数据

```
[XTrain,~ ,YTrain] = digitTrain4DArrayData; %加载训练图像样本
[XValidation,~,YValidation] = digitTest4DArrayData; %加载验证图像样本

% 随机显示20幅训练图像
numTrainImages = numel(YTrain); %统计用于训练样本的数量
figure
idx = randperm(numTrainImages,20);
for i = 1:numel(idx)
    subplot(4,5,i)
    imshow(XTrain(:, :, :, idx(i)))
    drawnow
end
```

步骤2:构建卷积神经网络

```
layers = [
    imageInputLayer([28 28 1]) % 输入层，1个通道，像素为28×28

    convolution2dLayer(3,8,'Padding','same') % 卷积层1：卷积核大小为3×3，卷积核的个数为8（每个卷积核的通道数与输入图像的通道数相同）
    batchNormalizationLayer % 批量归一化层1
    reluLayer % ReLU非线性激活函数1
    %averagePooling2dLayer(2,'Stride',2) % 池化层1：池化方式：平均池化；池化区域为2×2，步长为2
    maxPooling2dLayer(2,'Stride',2)

    convolution2dLayer(3,16,'Padding','same') % 卷积层2：卷积核大小为3×3，卷积核的个数为16（每个卷积核的通道数与输入特征图通道数相同）
    batchNormalizationLayer % 批量归一化层2
    reluLayer % ReLU非线性激活函数2
    %averagePooling2dLayer(2,'Stride',2) % 池化层2：池化方式：平均池化；池化区域为2×2，步长为2
    maxPooling2dLayer(2,'Stride',2)

    convolution2dLayer(3,32,'Padding','same') % 卷积层3：卷积核大小为3×3，卷积核的个数为32（每个卷积核的通道数与输入特征图通道数相同）
    batchNormalizationLayer % 批量归一化层3
    reluLayer % ReLU非线性激活函数3

    % convolution2dLayer(3,64,'Padding','same')
    % batchNormalizationLayer
    % reluLayer

    %dropoutLayer(0.2) % dropout层，随机将20%的输入置零
    fullyConnectedLayer(1) % 全连接层，全连接层的输出为1
    regressionLayer ]; % 回归层，用于预测结果
```

步骤3：配置训练选项

```
miniBatchSize = 128; % 训练一次最小的样本量为128
validationFrequency = floor(numel(YTrain)/miniBatchSize); % 验证频率
options = trainingOptions('sgdm', ... % 设置训练方法，本例中将其设置为SGDM法
    'MiniBatchSize',miniBatchSize, ... % 设置最小样本训练数量，本例中将其设置为128
    'MaxEpochs',30, ... % 设置最大训练轮数，在本例当中，最大训练轮数为30
    'InitialLearnRate',0.001, ... % 设置初始学习率为0.001
    'LearnRateSchedule','piecewise', ... % 设置初始的学习率是变化的
    'LearnRateDropFactor',0.1, ... % 设置学习率衰减因子为0.1
    'LearnRateDropPeriod',20, ... % 设置学习率衰减周期为20轮，即：每20轮，在之前的学习率基础上，乘以学习率的衰减
    'Shuffle','every-epoch', ... % 设置每一轮都打乱数据
    'ValidationData',{XValidation,YValidation}, ... % 设置验证用数据
    'ValidationFrequency',validationFrequency, ... % 设置验证频率
    'Plots','training-progress', ... % 设置打开训练进度图
    'Verbose',true); % 设置关闭命令窗口的输出
```

步骤4：训练网络

```
net = trainNetwork(XTrain,YTrain,layers,options);
```

步骤5：测试与评估

```
YPredicted = predict(net,XValidation); % 用训练好的网络预测验证图像中数字倾斜的角度
predictionError = YValidation - YPredicted; % 计算预测倾斜角度和实际倾斜角度之间的预测误差
```

```
% 计算准确率
thr = 10;
numCorrect = sum(abs(predictionError) < thr);
numValidationImages = numel(YValidation);
Accuracy = numCorrect/numValidationImages
% 计算RMSE的值
squares = predictionError.^2;
RMSE = sqrt(mean(squares))

% 设定阈值，在本例中，阈值设定为10度
% 当预测值与实际值得误差小于10度时，则认为预测正确
% 用于验证图像的数量
% 计算准确率
```