Contents

Introduction

This script uses matlab deep learning toolbox instead of implementing networks from scratch. The results that are of interest are discussed in the report.

Loading data

```
[xTrain, tTrain, xValid, tValid, xTest, tTest] = LoadCIFAR(3);



%Plot first 20 images in the training data
for i = 1:20
    subplot(4,5,i);
    imshow(xTrain(:,:,:,i))
end

error('Run the next sections to construct the models using deep learning toolbox. Note tha
t training can take hours. Deep learning toolbox in matlab allows users to use GPU effecti
ve computation')
```

```
ans =

          32          32           3       50000
```

```
Error using Rely_softmax_earlystopping_script (line 18)
Run the next sections to construct the models using deep learning toolbox. Note that train
ing can take hours. Deep learning toolbox in matlab allows users to use GPU effective comp
utation
```

Network 1

```
layers_net1 = [ ...
    imageInputLayer([32 32 3])
    fullyConnectedLayer(50)
    reluLayer
    fullyConnectedLayer(50)
    reluLayer
    fullyConnectedLayer(10)
    softmaxLayer
    classificationLayer];
```

```
options = trainingOptions('sgdm', ...
    'MaxEpochs',400,...
    'Shuffle','every-epoch', ...
    'MiniBatchSize',8192, ...
    'InitialLearnRate',1e-3, ...
    'Momentum',0.9, ...
    'ValidationData',{xValid, tValid}, ...
    'ValidationPatience',3, ...
    'ValidationFrequency',30, ...
    'Plots','training-progress');

[net_1, net1_info] = trainNetwork(xTrain,tTrain,layers_net1,options)

tPred = classify(net_1,xTest);
accuracy = sum(tPred == tTest)/numel(tTest)
```

Network 2

- Net 2 has more layers then net 1
- Net 2 has also a higher learning rate

```
layers_net2 = [ ...
    imageInputLayer([32 32 3])
    fullyConnectedLayer(50)
    reluLayer
    fullyConnectedLayer(50)
    reluLayer
    fullyConnectedLayer(50)
    reluLayer
    fullyConnectedLayer(10)
    softmaxLayer
    classificationLayer];

options = trainingOptions('sgdm', ...
    'MaxEpochs',400,...
    'Shuffle','every-epoch', ...
    'MiniBatchSize',8192, ...
    'InitialLearnRate',3e-3, ...
    'Momentum',0.9, ...
    'ValidationData',{xValid, tValid}, ...
    'ValidationPatience',3, ...
    'ValidationFrequency',30, ...
    'Plots','training-progress');

[net_2, net2_info] = trainNetwork(xTrain,tTrain,layers_net2,options)

tPred = classify(net_2,xTest);
accuracy = sum(tPred == tTest)/numel(tTest)
```

Network 3

- Net 3 has same layer layout as net 1 but includes regularization

```
options = trainingOptions('sgdm', ...
    'MaxEpochs',400,...
    'Shuffle','every-epoch', ...
    'MiniBatchSize',8192, ...
```

```
    'InitialLearnRate',1e-3, ...
    'L2Regularization',0.2, ...
    'ValidationData',{xValid, tValid}, ...
    'ValidationPatience',3, ...
    'ValidationFrequency',30, ...
    'Plots','training-progress');

[net_3, net3_info] = trainNetwork(xTrain,tTrain,layers_net1,options)

tPred = classify(net_3,xTest);
accuracy = sum(tPred == tTest)/numel(tTest)
```

Published with MATLAB® R2017b