Image Recognition

Datasets

I. POSE dataset

Cropped images of 68 subjects. Each subject has 13 poses.

Access the ith pose of the jth subject as: `pose(:, :, i, j)`

10 poses of each subject in training set and the rest 3 poses in the test set.

```
data = load('pose.mat');
% 68 subjects; 13 different poses per subject.
pose = data.pose; % 48x40x13x68
```

II. MNIST dataset

http://yann.lecun.com/exdb/mnist/

Training set has 60,000 examples, 28 x 28 grayscale images of handwritten digits (10 classes) and a testing set has 10,000 images.

```
data = load('mnist.mat');
mTrain = data.imgs_train;
mTest = data.imgs_test;
mTrainLabel = data.labels_train;
mTestLabel = data.labels_test;
```

Classification

I. Identifying subjects in the pose dataset

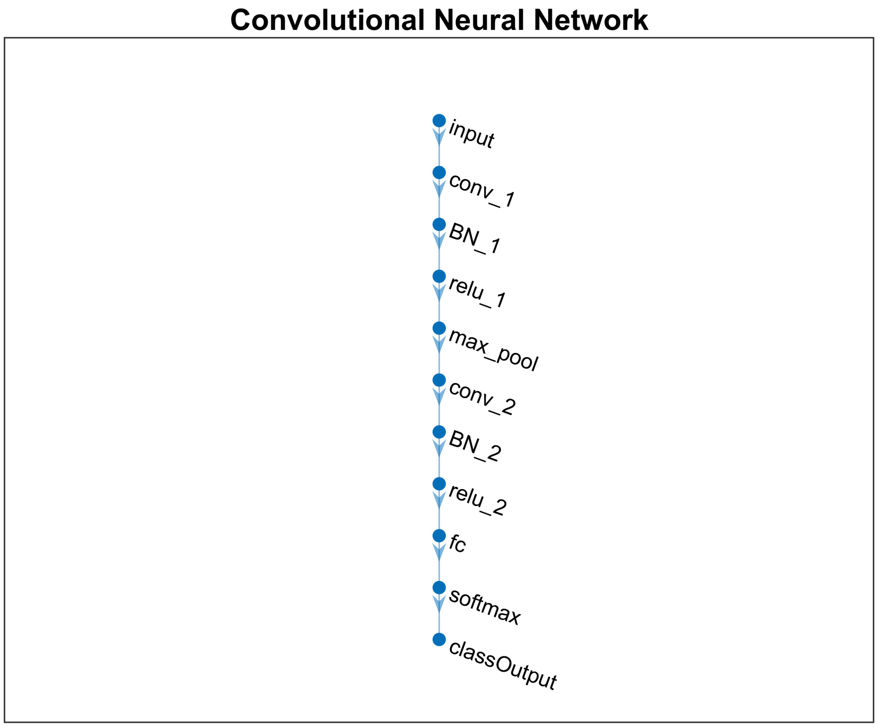
Creating train-test datasets and their corresponding labels

```
[d1,d2,npose,nsub] = size(pose);
trnnum = 10; % training set size
tstnum = npose-trnnum;  % test set size
% training set
pTrain = zeros(d1,d2,1,nsub*trnnum);
for i = 1:nsub
    count = (i-1)*trnnum;
    for j = 1:trnnum
        pTrain(:,:,1,count+j) = pose(:,:,j,i);
    end
end
pTrainLabel = categorical((kron(1:nsub,ones(1,trnnum)))');
% testing set
```

```
pTest = zeros(d1,d2,1,nsub*tstnum);
for i = 1:nsub
    count = (i-1)*tstnum;
    for j = 1:tstnum
        pTest(:,:,1,count+j) = pose(:,:,trnnum+j,i);
    end
end
pTestLabel = categorical((kron(1:nsub,ones(1,tstnum)))');
```

Defining a Neural Network

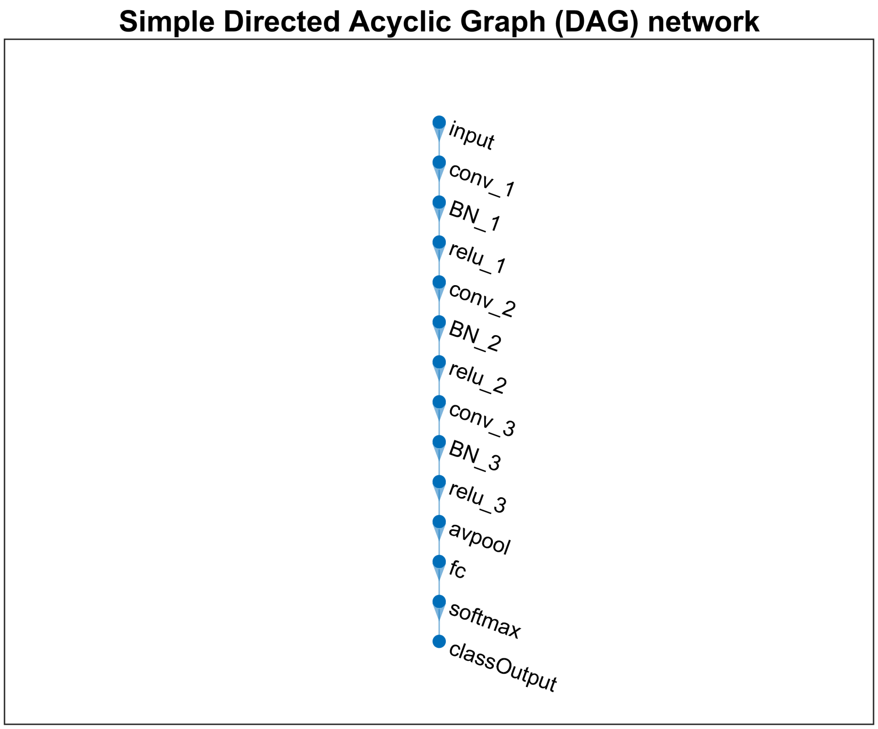
```
%------------------------------------------------------------------------
% Convolutional neural network
layers = [
    imageInputLayer([d1,d2,1],'Name','input')
    convolution2dLayer(3,16,'Padding',1,'Name','conv_1')
    batchNormalizationLayer('Name','BN_1')
    reluLayer('Name','relu_1')
    maxPooling2dLayer(2,'Stride',2,'Name','max_pool')
    convolution2dLayer(3,32,'Padding',1,'Name','conv_2')
    batchNormalizationLayer('Name','BN_2')
    reluLayer('Name','relu_2')
    fullyConnectedLayer(nsub,'Name','fc')
    softmaxLayer('Name','softmax')
    classificationLayer('Name','classOutput')];

lgraph = layerGraph(layers);
figure
plot(lgraph)
title('Convolutional Neural Network')

options = trainingOptions('sgdm', ...
    'MaxEpochs',20,...
    'InitialLearnRate',1e-3, ...
    'Verbose',false, ...
    'Plots','training-progress');

net = trainNetwork(pTrain,pTrainLabel,layers,options);
```



```
analyzeNetwork(net)

% Classifying test data
pred_test_labels = classify(net,pTest);
% numel returns the number of elements in the array
accuracy = sum(pred_test_labels == pTestLabel)/numel(pTestLabel);
fprintf("accuracy is %f%",accuracy*100);
```

```
accuracy is 60.294118
```

```

%------------------------------------------------------------------------
% Simple Directed Acyclic Graph (DAG) network
layers = [
    imageInputLayer([d1,d2,1],'Name','input')

    convolution2dLayer(5,16,'Padding','same','Name','conv_1')
    batchNormalizationLayer('Name','BN_1')
    reluLayer('Name','relu_1')

    convolution2dLayer(3,32,'Padding','same','Stride',2,'Name','conv_2')
    batchNormalizationLayer('Name','BN_2')
    reluLayer('Name','relu_2')
    convolution2dLayer(3,32,'Padding','same','Name','conv_3')
    batchNormalizationLayer('Name','BN_3')
    reluLayer('Name','relu_3')

    %additionLayer(2,'Name','add')
    averagePooling2dLayer(2,'Stride',2,'Name','avpool')
    fullyConnectedLayer(nsub,'Name','fc')
    softmaxLayer('Name','softmax')
    classificationLayer('Name','classOutput')];

lgraph = layerGraph(layers);
figure
plot(lgraph)
title('Simple Directed Acyclic Graph (DAG) network')

options = trainingOptions('sgdm', ...
    'MaxEpochs',15,...
    'Shuffle','every-epoch', ...
    'Verbose',false, ...
    'Plots','training-progress');

net = trainNetwork(pTrain,pTrainLabel,layers,options);
```



Simple Directed Acyclic Graph (DAG) network

```
analyzeNetwork(net)

% Classifying test data
pred_test_labels = classify(net,pTest);
% numel returns the number of elements in the array
accuracy = sum(pred_test_labels == pTestLabel)/numel(pTestLabel);
fprintf("accuracy is %f%",accuracy*100);
```

```
accuracy is 65.196078
```

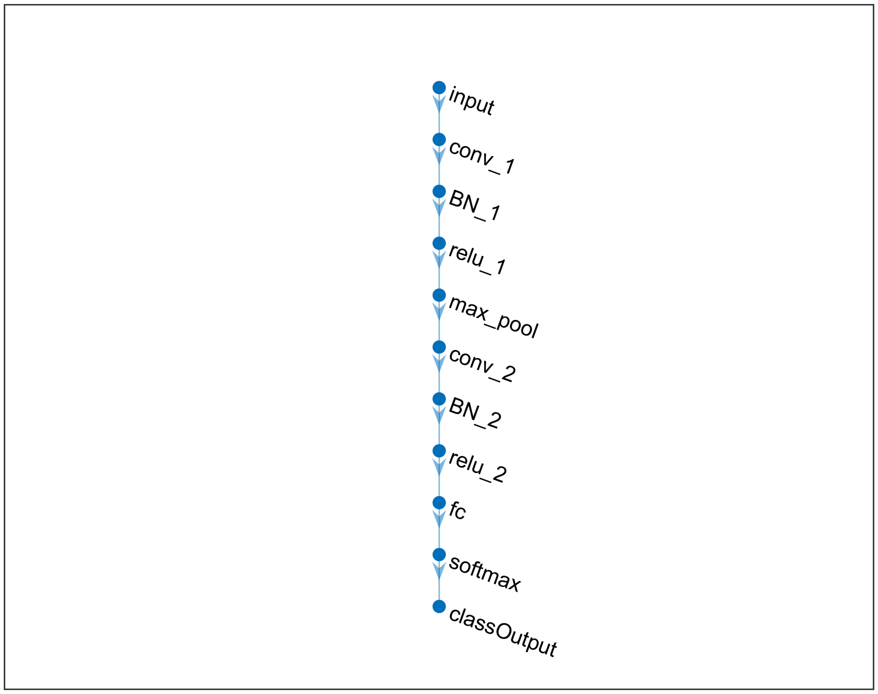
II. Identifying handwritten digits in the MNIST dataset

Train-test datasets and their corresponding labels

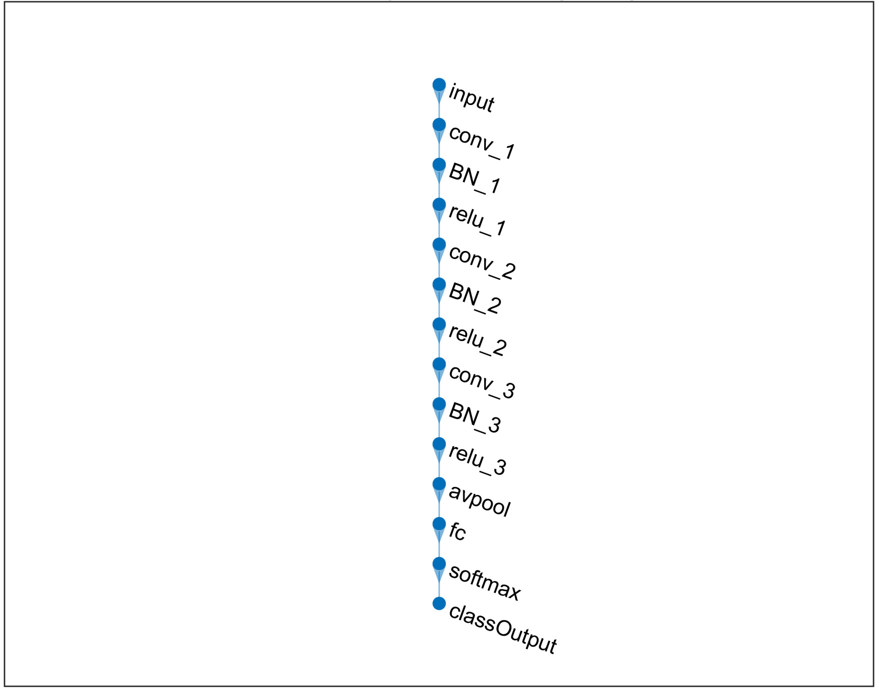
```
[d1,d2,trnnum] = size(mTrain);
tstnum = size(mTest,3);
train = zeros(d1,d2,1,trnnum);
test = zeros(d1,d2,1,tstnum);
train(:,:,1,:) = mTrain;
test(:,:,1,:) = mTest;
```

Defining a Neural Network

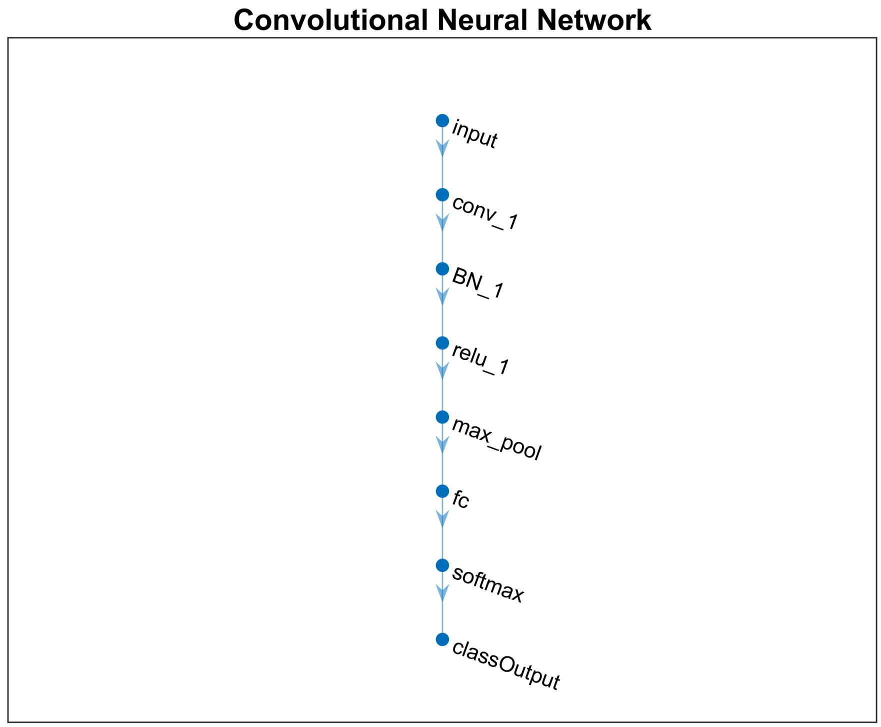
```
%-----------------------------------------------------------------------
% Convolutional neural network
layers = [ ...
    imageInputLayer([d1,d2,1],'Name','input')
    convolution2dLayer(3,16,'Name','conv_1')
    batchNormalizationLayer('Name','BN_1')
    reluLayer('Name','relu_1')
    maxPooling2dLayer(2,'Stride',2,'Name','max_pool')
    fullyConnectedLayer(10,'Name','fc')
    softmaxLayer('Name','softmax')
    classificationLayer('Name','classOutput')];

lgraph = layerGraph(layers);
figure
plot(lgraph)
title('Convolutional Neural Network')

options = trainingOptions('sgdm', ...
    'MaxEpochs',7,...
    'InitialLearnRate',1e-3, ...
    'Verbose',false, ...
    'Plots','training-progress');

net = trainNetwork(train,mTrainLabel,layers,options);
```



Convolutional Neural Network



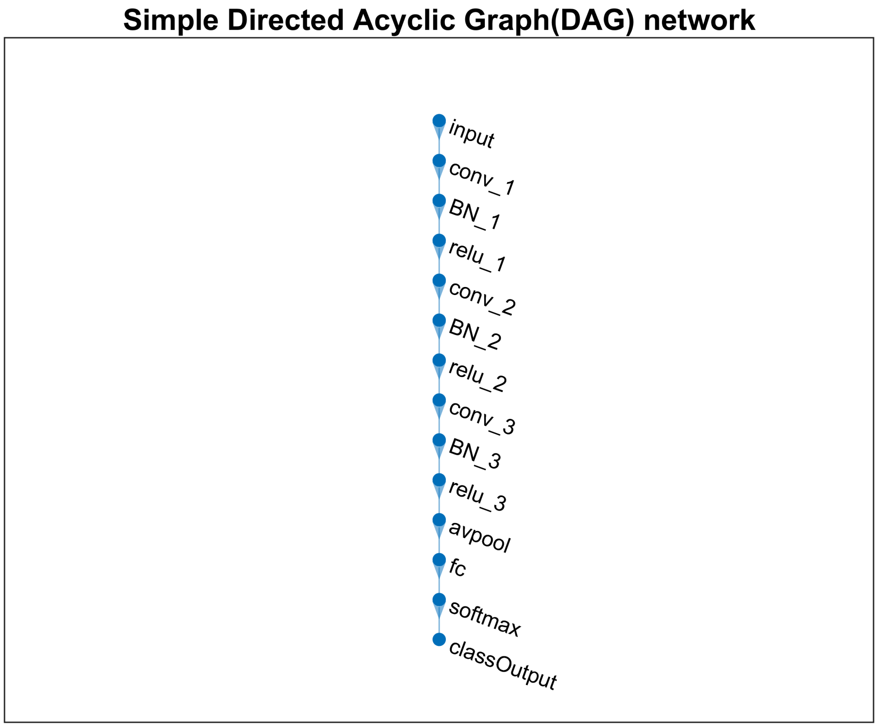
Simple Directed Acyclic Graph (DAG) network



```
analyzeNetwork(net)

% Classifying test data
pred_test_labels = classify(net,test);
% numel returns the number of elements in the array
accuracy = sum(pred_test_labels == mTestLabel)/numel(mTestLabel);
fprintf("accuracy is %f%",accuracy*100);
```

```
accuracy is 97.030000
```

```

%----------------------------------------------------------------------
% Simple Directed Acyclic Graph (DAG) network
layers = [
    imageInputLayer([d1,d2,1],'Name','input')

    convolution2dLayer(5,16,'Padding','same','Name','conv_1')
    batchNormalizationLayer('Name','BN_1')
    reluLayer('Name','relu_1')

    convolution2dLayer(3,32,'Padding','same','Stride',2,'Name','conv_2')
    batchNormalizationLayer('Name','BN_2')
    reluLayer('Name','relu_2')
    convolution2dLayer(3,32,'Padding','same','Name','conv_3')
    batchNormalizationLayer('Name','BN_3')
    reluLayer('Name','relu_3')

    %additionLayer(2,'Name','add')
    averagePooling2dLayer(2,'Stride',2,'Name','avpool')
    fullyConnectedLayer(10,'Name','fc')
    softmaxLayer('Name','softmax')
    classificationLayer('Name','classOutput')];

lgraph = layerGraph(layers);
figure
plot(lgraph)
title('Simple Directed Acyclic Graph(DAG) network')

options = trainingOptions('sgdm', ...
    'MaxEpochs',7,...
    'Shuffle','every-epoch', ...
    'Verbose',false, ...
    'Plots','training-progress');

net = trainNetwork(train,mTrainLabel,layers,options);
```



```
analyzeNetwork(net)

% Classifying test data
pred_test_labels = classify(net,test);
% numel returns the number of elements in the array
accuracy = sum(pred_test_labels == mTestLabel)/numel(mTestLabel);
fprintf("accuracy is %f%",accuracy*100);
```

```
accuracy is 99.000000
```