Create Deep Learning Network Architecture

Script for creating the layers for a deep learning network with the following properties:

```
Number of layers: 144
Number of connections: 170
```

Run the script to create the layers in the workspace variable `lgraph`.

To learn more, see Generate MATLAB Code From Deep Network Designer.

Auto-generated by MATLAB on 19-May-2023 18:05:50

Create Layer Graph

Create the layer graph variable to contain the network layers.

```
lgraph = layerGraph();
```

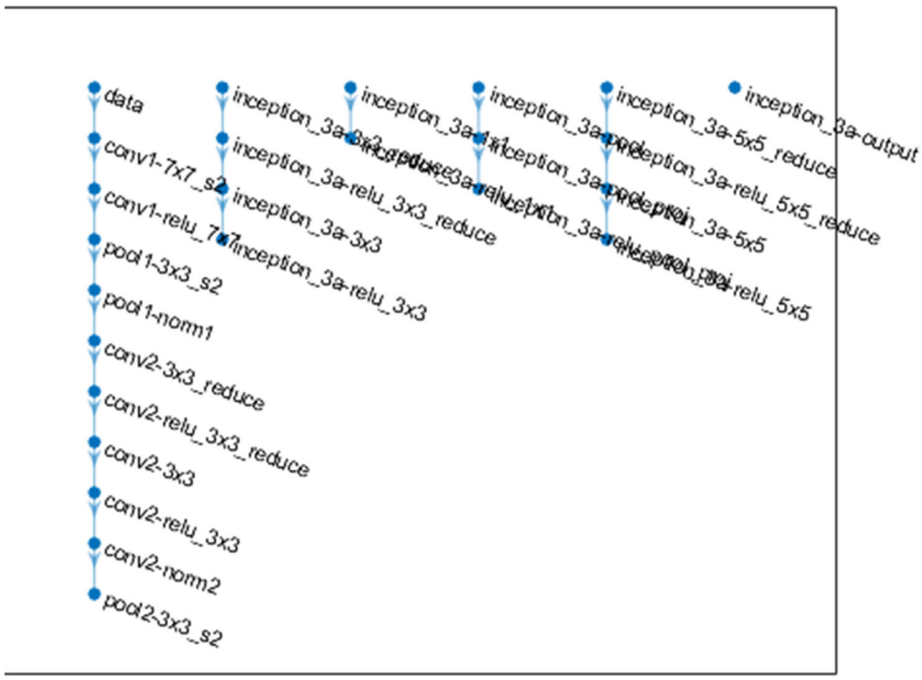
Add Layer Branches

Add the branches of the plot network to the layer graph. Input Layer → 2 Each branch is a linear array of layers.

```
tempLayers = [
% Input Layer with convolution [ 7 7 ]
    imageInputLayer([224 224 3],"Name","data")

    convolution2dLayer([7 7],64,"Name","conv1-
7x7_s2","BiasLearnRateFactor",2,"Padding",[3 3 3 3],"Stride",[2 2])
    reluLayer("Name","conv1-relu_7x7")
    maxPooling2dLayer([3 3],"Name","pool1-3x3_s2","Padding",[0 1 0
1],"Stride",[2 2])
    crossChannelNormalizationLayer(5,"Name","pool1-norm1","K",1)
% Feature extraction  Layer with convolution [ 1 1 ] & [3 3]
    convolution2dLayer([1 1],64,"Name","conv2-
3x3_reduce","BiasLearnRateFactor",2)
    reluLayer("Name","conv2-relu_3x3_reduce")
    convolution2dLayer([3 3],192,"Name","conv2-
3x3","BiasLearnRateFactor",2,"Padding",[1 1 1 1])
    reluLayer("Name","conv2-relu_3x3")
    crossChannelNormalizationLayer(5,"Name","conv2-norm2","K",1)
    maxPooling2dLayer([3 3],"Name","pool2-3x3_s2","Padding",[0 1 0
1],"Stride",[2 2])];

lgraph = addLayers(lgraph,tempLayers);
```

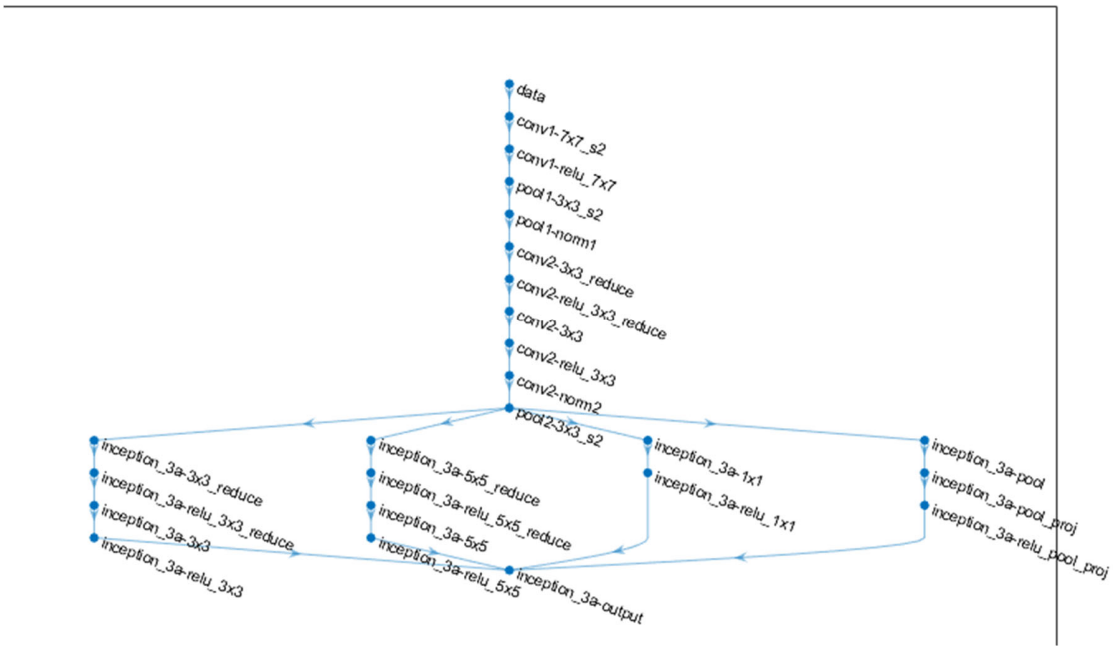

```
 tempLayers = [
     convolution2dLayer([1 1],96,"Name","inception_3a-
3x3_reduce","BiasLearnRateFactor",2)
     reluLayer("Name","inception_3a-relu_3x3_reduce")
     convolution2dLayer([3 3],128,"Name","inception_3a-
3x3","BiasLearnRateFactor",2,"Padding",[1 1 1 1])
     reluLayer("Name","inception_3a-relu_3x3")];
 lgraph = addLayers(lgraph,tempLayers);

 tempLayers = [
     convolution2dLayer([1 1],64,"Name","inception_3a-
1x1","BiasLearnRateFactor",2)
     reluLayer("Name","inception_3a-relu_1x1")];
 lgraph = addLayers(lgraph,tempLayers);

 tempLayers = [
     maxPooling2dLayer([3 3],"Name","inception_3a-pool","Padding",[1 1 1 1])
     convolution2dLayer([1 1],32,"Name","inception_3a-
pool_proj","BiasLearnRateFactor",2)
     reluLayer("Name","inception_3a-relu_pool_proj")];
 lgraph = addLayers(lgraph,tempLayers);

 tempLayers = [
     convolution2dLayer([1 1],16,"Name","inception_3a-
5x5_reduce","BiasLearnRateFactor",2)
     reluLayer("Name","inception_3a-relu_5x5_reduce")
     convolution2dLayer([5 5],32,"Name","inception_3a-
5x5","BiasLearnRateFactor",2,"Padding",[2 2 2 2])
     reluLayer("Name","inception_3a-relu_5x5")];
 lgraph = addLayers(lgraph,tempLayers);
```

```
tempLayers = depthConcatenationLayer(4,"Name","inception_3a-output");
lgraph = addLayers(lgraph,tempLayers);
```



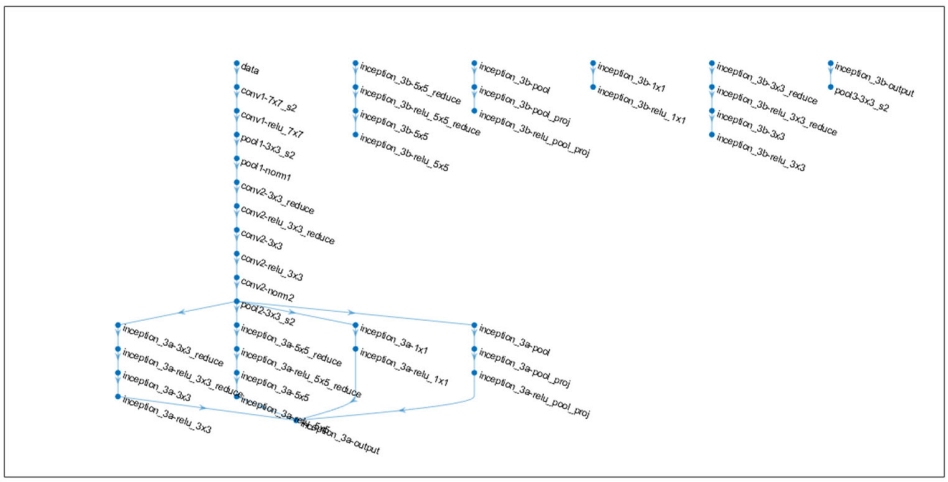

```
lgraph = connectLayers(lgraph,"pool2-3x3_s2","inception_3a-3x3_reduce");
lgraph = connectLayers(lgraph,"pool2-3x3_s2","inception_3a-1x1");
lgraph = connectLayers(lgraph,"pool2-3x3_s2","inception_3a-pool");
lgraph = connectLayers(lgraph,"pool2-3x3_s2","inception_3a-5x5_reduce");
lgraph = connectLayers(lgraph,"inception_3a-relu_1x1","inception_3a-
output/in1");
lgraph = connectLayers(lgraph,"inception_3a-relu_3x3","inception_3a-
output/in2");
lgraph = connectLayers(lgraph,"inception_3a-relu_pool_proj","inception_3a-
output/in4");
```

lgraph = connectLayers(lgraph,"inception_3a-relu_5x5","inception_3a-output/in3");

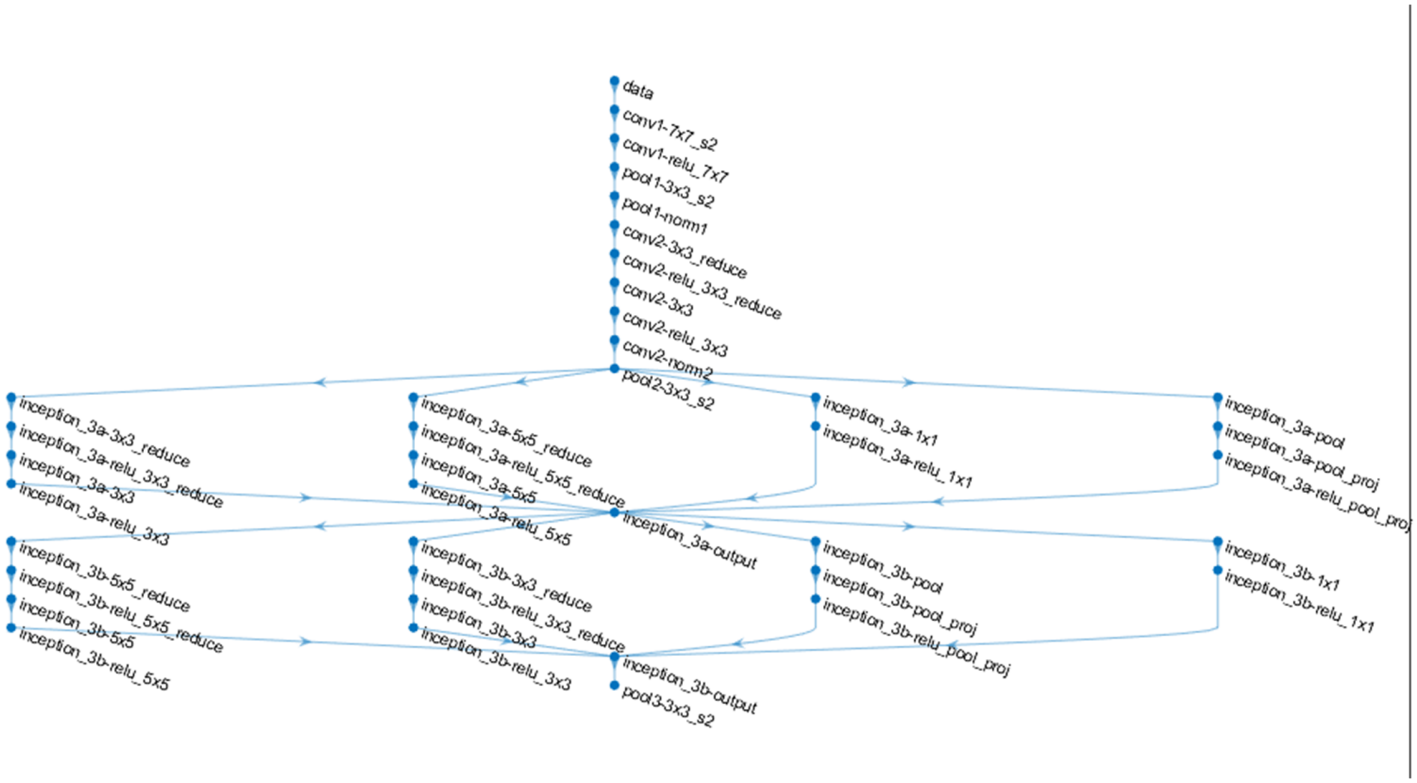



```
 tempLayers = [
     convolution2dLayer([1 1],32,"Name","inception_3b-
5x5_reduce","BiasLearnRateFactor",2)
     reluLayer("Name","inception_3b-relu_5x5_reduce")
     convolution2dLayer([5 5],96,"Name","inception_3b-
5x5","BiasLearnRateFactor",2,"Padding",[2 2 2 2])
     reluLayer("Name","inception_3b-relu_5x5")];
 lgraph = addLayers(lgraph,tempLayers);

 tempLayers = [
     maxPooling2dLayer([3 3],"Name","inception_3b-pool","Padding",[1 1 1 1])
     convolution2dLayer([1 1],64,"Name","inception_3b-
pool_proj","BiasLearnRateFactor",2)
     reluLayer("Name","inception_3b-relu_pool_proj")];
 lgraph = addLayers(lgraph,tempLayers);

 tempLayers = [
     convolution2dLayer([1 1],128,"Name","inception_3b-
1x1","BiasLearnRateFactor",2)
     reluLayer("Name","inception_3b-relu_1x1")];
 lgraph = addLayers(lgraph,tempLayers);

 tempLayers = [
     convolution2dLayer([1 1],128,"Name","inception_3b-
3x3_reduce","BiasLearnRateFactor",2)
     reluLayer("Name","inception_3b-relu_3x3_reduce")
```

```
    convolution2dLayer([3 3],192,"Name","inception_3b-
3x3","BiasLearnRateFactor",2,"Padding",[1 1 1 1])
    reluLayer("Name","inception_3b-relu_3x3")];
 lgraph = addLayers(lgraph,tempLayers);

 tempLayers = [
    depthConcatenationLayer(4,"Name","inception_3b-output")
    maxPooling2dLayer([3 3],"Name","pool3-3x3_s2","Padding",[0 1 0
1],"Stride",[2 2])];
 lgraph = addLayers(lgraph,tempLayers);
```

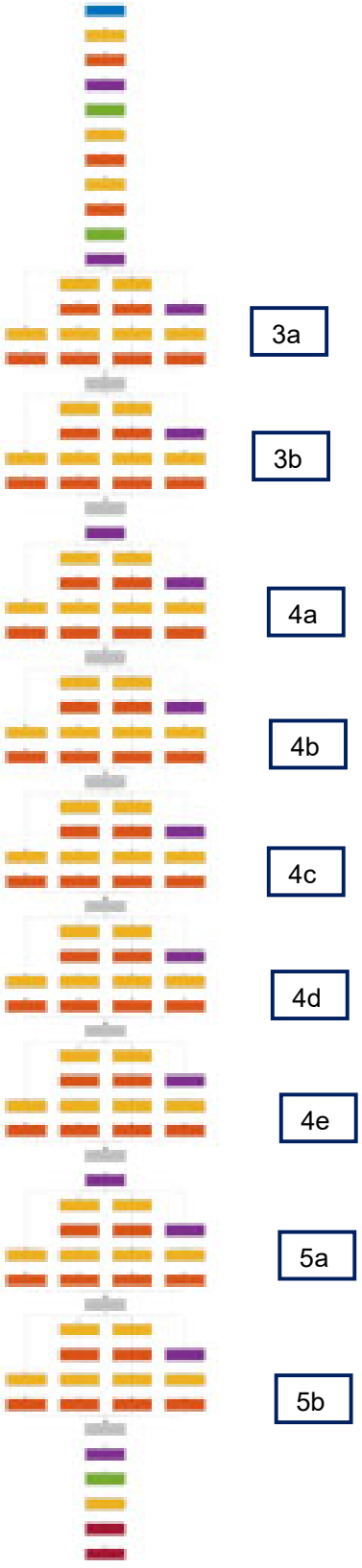


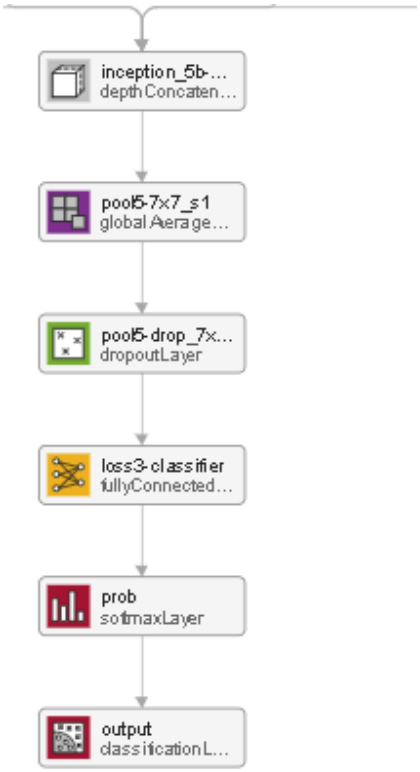

```
 lgraph = connectLayers(lgraph,"inception_3a-output","inception_3b-5x5_reduce");
 lgraph = connectLayers(lgraph,"inception_3a-output","inception_3b-pool");
 lgraph = connectLayers(lgraph,"inception_3a-output","inception_3b-1x1");
 lgraph = connectLayers(lgraph,"inception_3a-output","inception_3b-3x3_reduce");
 lgraph = connectLayers(lgraph,"inception_3b-relu_pool_proj","inception_3b-
output/in4");
 lgraph = connectLayers(lgraph,"inception_3b-relu_5x5","inception_3b-
output/in3");
 lgraph = connectLayers(lgraph,"inception_3b-relu_1x1","inception_3b-
output/in1");
 lgraph = connectLayers(lgraph,"inception_3b-relu_3x3","inception_3b-
output/in2");
```



Output layers

```
tempLayers = [
    depthConcatenationLayer(4,"Name","inception_5b-output")
    globalAveragePooling2dLayer("Name","pool5-7x7_s1")
    dropoutLayer(0.4,"Name","pool5-drop_7x7_s1")
    fullyConnectedLayer(1000,"Name","loss3-classifier","BiasLearnRateFactor",2)
    softmaxLayer("Name","prob")
    classificationLayer("Name","output")];
lgraph = addLayers(lgraph,tempLayers);
```


3a
3b
4a
4b
4c
4d
4e
5a
5b


inception_5b-...
depthConcaten...
pool5-7x7_s1
globalAverage...
pool5-drop_7x...
dropoutLayer
loss3-classifier
fullyConnected...
prob
softmaxLayer
output
classificationL...