Create Deep Learning Network Architecture

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

Number of layers: 71

Number of connections: 78

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

To learn more, see [Generate MATLAB Code From Deep Network Designer](matlab:helpview('deeplearning','generate_matlab_code')).

Auto-generated by MATLAB on 08-Apr-2023 14:10:33

# Create Layer Graph

Create the layer graph variable to contain the network layers.

lgraph = layerGraph();

# Add Layer Branches

Add the branches of the network to the layer graph. Each branch is a linear array of layers.

tempLayers = [

imageInputLayer([224 224 3],"Name","data","Normalization","zscore")

convolution2dLayer([7 7],64,"Name","conv1","BiasLearnRateFactor",0,"Padding",[3 3 3 3],"Stride",[2 2])

batchNormalizationLayer("Name","bn\_conv1")

reluLayer("Name","conv1\_relu")

maxPooling2dLayer([3 3],"Name","pool1","Padding",[1 1 1 1],"Stride",[2 2])];

lgraph = addLayers(lgraph,tempLayers);

tempLayers = [

convolution2dLayer([3 3],64,"Name","res2a\_branch2a","BiasLearnRateFactor",0,"Padding",[1 1 1 1])

batchNormalizationLayer("Name","bn2a\_branch2a")

reluLayer("Name","res2a\_branch2a\_relu")

convolution2dLayer([3 3],64,"Name","res2a\_branch2b","BiasLearnRateFactor",0,"Padding",[1 1 1 1])

batchNormalizationLayer("Name","bn2a\_branch2b")];

lgraph = addLayers(lgraph,tempLayers);

tempLayers = [

additionLayer(2,"Name","res2a")

reluLayer("Name","res2a\_relu")];

lgraph = addLayers(lgraph,tempLayers);

tempLayers = [

convolution2dLayer([3 3],64,"Name","res2b\_branch2a","BiasLearnRateFactor",0,"Padding",[1 1 1 1])

batchNormalizationLayer("Name","bn2b\_branch2a")

reluLayer("Name","res2b\_branch2a\_relu")

convolution2dLayer([3 3],64,"Name","res2b\_branch2b","BiasLearnRateFactor",0,"Padding",[1 1 1 1])

batchNormalizationLayer("Name","bn2b\_branch2b")];

lgraph = addLayers(lgraph,tempLayers);

tempLayers = [

additionLayer(2,"Name","res2b")

reluLayer("Name","res2b\_relu")];

lgraph = addLayers(lgraph,tempLayers);

tempLayers = [

convolution2dLayer([3 3],128,"Name","res3a\_branch2a","BiasLearnRateFactor",0,"Padding",[1 1 1 1],"Stride",[2 2])

batchNormalizationLayer("Name","bn3a\_branch2a")

reluLayer("Name","res3a\_branch2a\_relu")

convolution2dLayer([3 3],128,"Name","res3a\_branch2b","BiasLearnRateFactor",0,"Padding",[1 1 1 1])

batchNormalizationLayer("Name","bn3a\_branch2b")];

lgraph = addLayers(lgraph,tempLayers);

tempLayers = [

convolution2dLayer([1 1],128,"Name","res3a\_branch1","BiasLearnRateFactor",0,"Stride",[2 2])

batchNormalizationLayer("Name","bn3a\_branch1")];

lgraph = addLayers(lgraph,tempLayers);

tempLayers = [

additionLayer(2,"Name","res3a")

reluLayer("Name","res3a\_relu")];

lgraph = addLayers(lgraph,tempLayers);

tempLayers = [

convolution2dLayer([3 3],128,"Name","res3b\_branch2a","BiasLearnRateFactor",0,"Padding",[1 1 1 1])

batchNormalizationLayer("Name","bn3b\_branch2a")

reluLayer("Name","res3b\_branch2a\_relu")

convolution2dLayer([3 3],128,"Name","res3b\_branch2b","BiasLearnRateFactor",0,"Padding",[1 1 1 1])

batchNormalizationLayer("Name","bn3b\_branch2b")];

lgraph = addLayers(lgraph,tempLayers);

tempLayers = [

additionLayer(2,"Name","res3b")

reluLayer("Name","res3b\_relu")];

lgraph = addLayers(lgraph,tempLayers);

tempLayers = [

convolution2dLayer([3 3],256,"Name","res4a\_branch2a","BiasLearnRateFactor",0,"Padding",[1 1 1 1],"Stride",[2 2])

batchNormalizationLayer("Name","bn4a\_branch2a")

reluLayer("Name","res4a\_branch2a\_relu")

convolution2dLayer([3 3],256,"Name","res4a\_branch2b","BiasLearnRateFactor",0,"Padding",[1 1 1 1])

batchNormalizationLayer("Name","bn4a\_branch2b")];

lgraph = addLayers(lgraph,tempLayers);

tempLayers = [

convolution2dLayer([1 1],256,"Name","res4a\_branch1","BiasLearnRateFactor",0,"Stride",[2 2])

batchNormalizationLayer("Name","bn4a\_branch1")];

lgraph = addLayers(lgraph,tempLayers);

tempLayers = [

additionLayer(2,"Name","res4a")

reluLayer("Name","res4a\_relu")];

lgraph = addLayers(lgraph,tempLayers);

tempLayers = [

convolution2dLayer([3 3],256,"Name","res4b\_branch2a","BiasLearnRateFactor",0,"Padding",[1 1 1 1])

batchNormalizationLayer("Name","bn4b\_branch2a")

reluLayer("Name","res4b\_branch2a\_relu")

convolution2dLayer([3 3],256,"Name","res4b\_branch2b","BiasLearnRateFactor",0,"Padding",[1 1 1 1])

batchNormalizationLayer("Name","bn4b\_branch2b")];

lgraph = addLayers(lgraph,tempLayers);

tempLayers = [

additionLayer(2,"Name","res4b")

reluLayer("Name","res4b\_relu")];

lgraph = addLayers(lgraph,tempLayers);

tempLayers = [

convolution2dLayer([1 1],512,"Name","res5a\_branch1","BiasLearnRateFactor",0,"Stride",[2 2])

batchNormalizationLayer("Name","bn5a\_branch1")];

lgraph = addLayers(lgraph,tempLayers);

tempLayers = [

convolution2dLayer([3 3],512,"Name","res5a\_branch2a","BiasLearnRateFactor",0,"Padding",[1 1 1 1],"Stride",[2 2])

batchNormalizationLayer("Name","bn5a\_branch2a")

reluLayer("Name","res5a\_branch2a\_relu")

convolution2dLayer([3 3],512,"Name","res5a\_branch2b","BiasLearnRateFactor",0,"Padding",[1 1 1 1])

batchNormalizationLayer("Name","bn5a\_branch2b")];

lgraph = addLayers(lgraph,tempLayers);

tempLayers = [

additionLayer(2,"Name","res5a")

reluLayer("Name","res5a\_relu")];

lgraph = addLayers(lgraph,tempLayers);

tempLayers = [

convolution2dLayer([3 3],512,"Name","res5b\_branch2a","BiasLearnRateFactor",0,"Padding",[1 1 1 1])

batchNormalizationLayer("Name","bn5b\_branch2a")

reluLayer("Name","res5b\_branch2a\_relu")

convolution2dLayer([3 3],512,"Name","res5b\_branch2b","BiasLearnRateFactor",0,"Padding",[1 1 1 1])

batchNormalizationLayer("Name","bn5b\_branch2b")];

lgraph = addLayers(lgraph,tempLayers);

tempLayers = [

additionLayer(2,"Name","res5b")

reluLayer("Name","res5b\_relu")

globalAveragePooling2dLayer("Name","pool5")

fullyConnectedLayer(1000,"Name","fc1000")

softmaxLayer("Name","prob")

classificationLayer("Name","ClassificationLayer\_predictions")];

lgraph = addLayers(lgraph,tempLayers);

% clean up helper variable

clear tempLayers;

# Connect Layer Branches

Connect all the branches of the network to create the network graph.

lgraph = connectLayers(lgraph,"pool1","res2a\_branch2a");

lgraph = connectLayers(lgraph,"pool1","res2a/in2");

lgraph = connectLayers(lgraph,"bn2a\_branch2b","res2a/in1");

lgraph = connectLayers(lgraph,"res2a\_relu","res2b\_branch2a");

lgraph = connectLayers(lgraph,"res2a\_relu","res2b/in2");

lgraph = connectLayers(lgraph,"bn2b\_branch2b","res2b/in1");

lgraph = connectLayers(lgraph,"res2b\_relu","res3a\_branch2a");

lgraph = connectLayers(lgraph,"res2b\_relu","res3a\_branch1");

lgraph = connectLayers(lgraph,"bn3a\_branch1","res3a/in2");

lgraph = connectLayers(lgraph,"bn3a\_branch2b","res3a/in1");

lgraph = connectLayers(lgraph,"res3a\_relu","res3b\_branch2a");

lgraph = connectLayers(lgraph,"res3a\_relu","res3b/in2");

lgraph = connectLayers(lgraph,"bn3b\_branch2b","res3b/in1");

lgraph = connectLayers(lgraph,"res3b\_relu","res4a\_branch2a");

lgraph = connectLayers(lgraph,"res3b\_relu","res4a\_branch1");

lgraph = connectLayers(lgraph,"bn4a\_branch1","res4a/in2");

lgraph = connectLayers(lgraph,"bn4a\_branch2b","res4a/in1");

lgraph = connectLayers(lgraph,"res4a\_relu","res4b\_branch2a");

lgraph = connectLayers(lgraph,"res4a\_relu","res4b/in2");

lgraph = connectLayers(lgraph,"bn4b\_branch2b","res4b/in1");

lgraph = connectLayers(lgraph,"res4b\_relu","res5a\_branch1");

lgraph = connectLayers(lgraph,"res4b\_relu","res5a\_branch2a");

lgraph = connectLayers(lgraph,"bn5a\_branch1","res5a/in2");

lgraph = connectLayers(lgraph,"bn5a\_branch2b","res5a/in1");

lgraph = connectLayers(lgraph,"res5a\_relu","res5b\_branch2a");

lgraph = connectLayers(lgraph,"res5a\_relu","res5b/in2");

lgraph = connectLayers(lgraph,"bn5b\_branch2b","res5b/in1");

# Plot Layers

plot(lgraph);