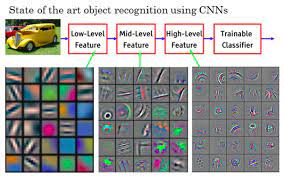
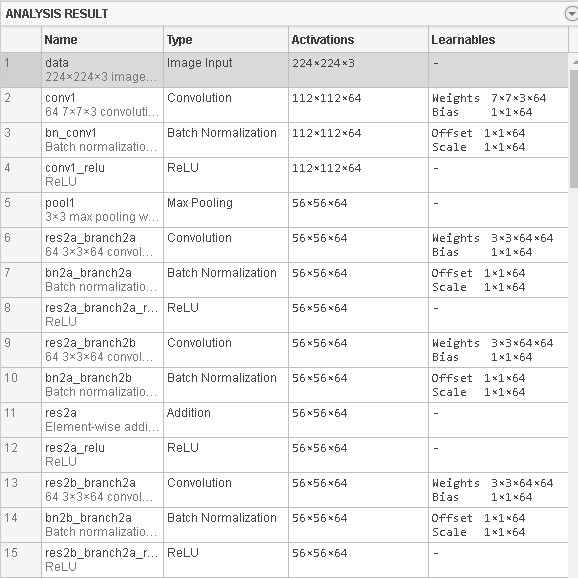
5.2-2 ResNet -- Create Deep Learning Network Architecture- ResNet18

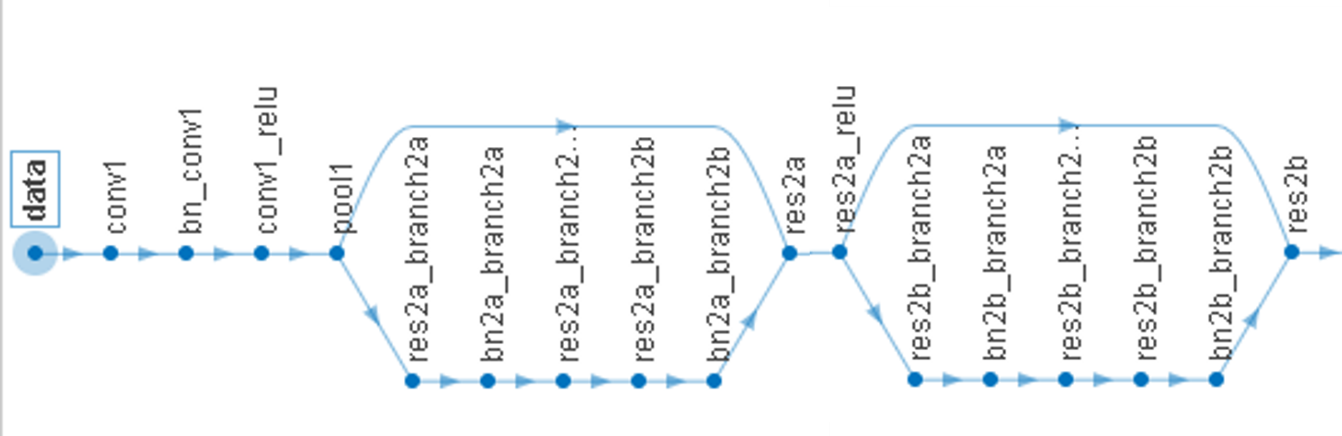
>> deepNetworkDesigner

Import ResNet18 & Analyze

Input layers 🡪 Feature extraction (Convolution) Layers 🡪 FC layers 🡪 output Layers







1st feature extraction

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.

# The Input layers & first branch res2b

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);

# Connect Layer Branches

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

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

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

# The second branch res2b

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")];

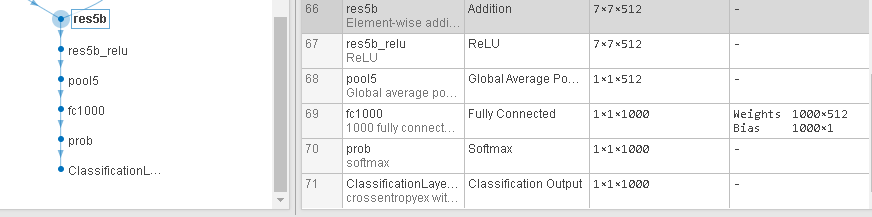
lgraph = addLayers(lgraph,tempLayers);

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

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

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

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

FC layers & output layers

# 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);

完整的程式請參考 ResNet18.docx