Convolutional Neural Networks

範例與APP使用

MALAB進階程式語言與實作

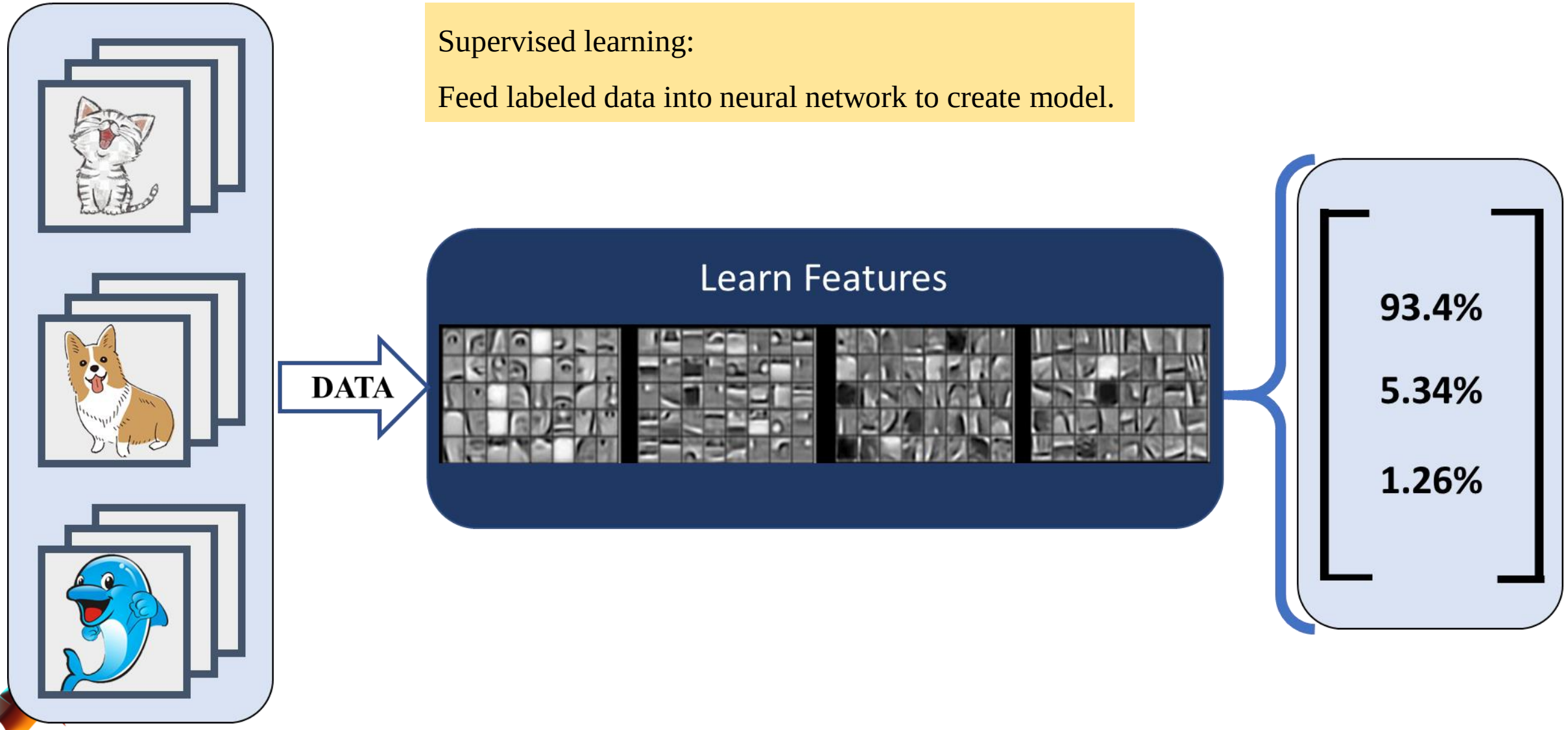
生醫光電所 黃梓軒

What is Deep Learning?

Deep learning is a type of machine learning based on ***artificial neural networks*** with features learning ***directly from data.***

Supervised learning:

Feed labeled data into neural network to create model.



Deep learning task using MATLAB

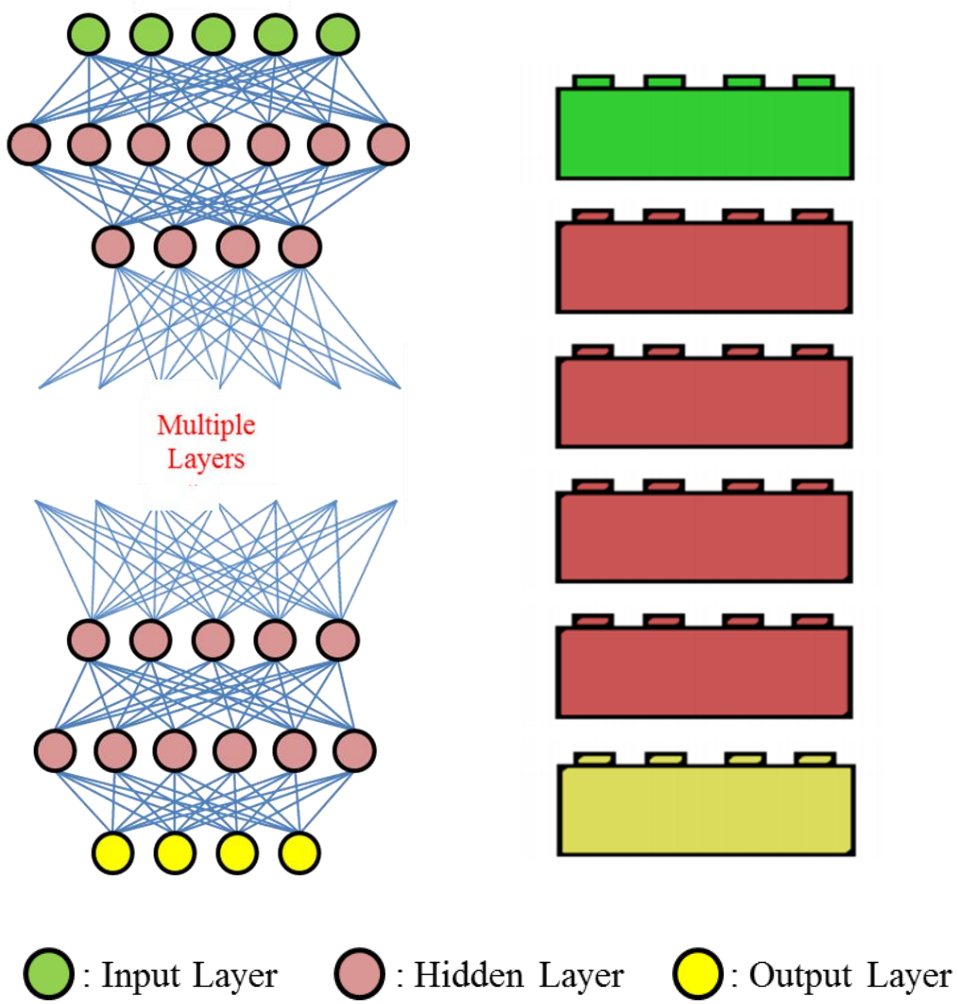
- Building Convolution Neural Network

- Data Preparing

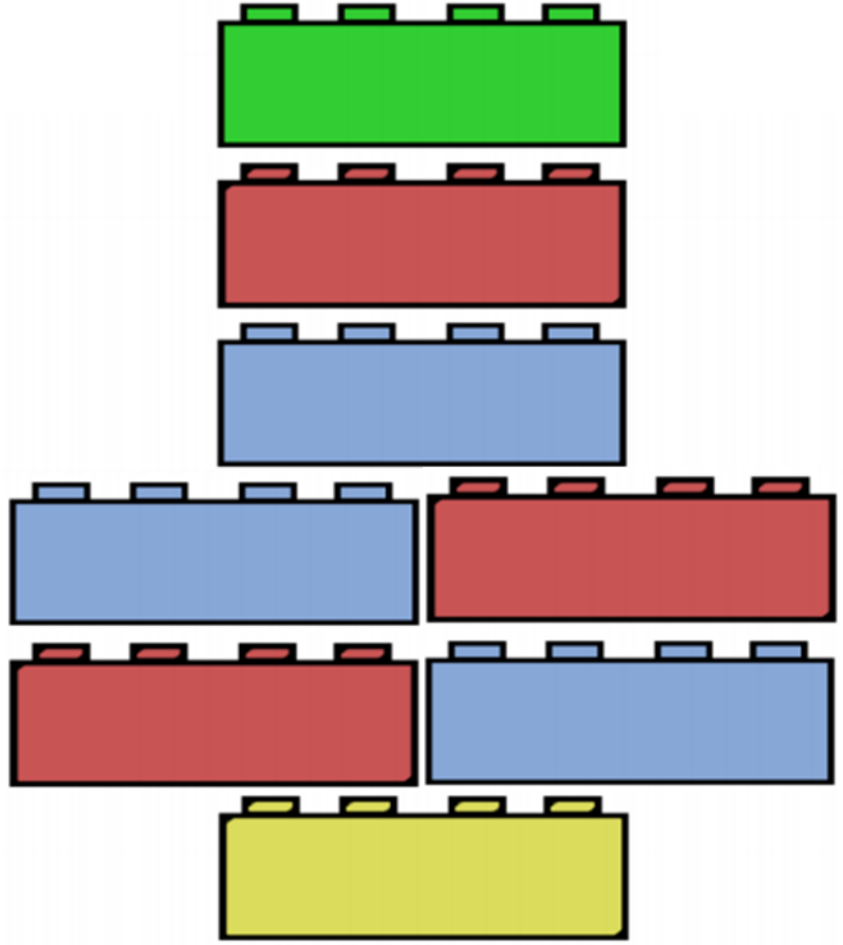
- HyperParameter Setting

Artificial Neural Networks

Layers are like blocks, stack on each other.



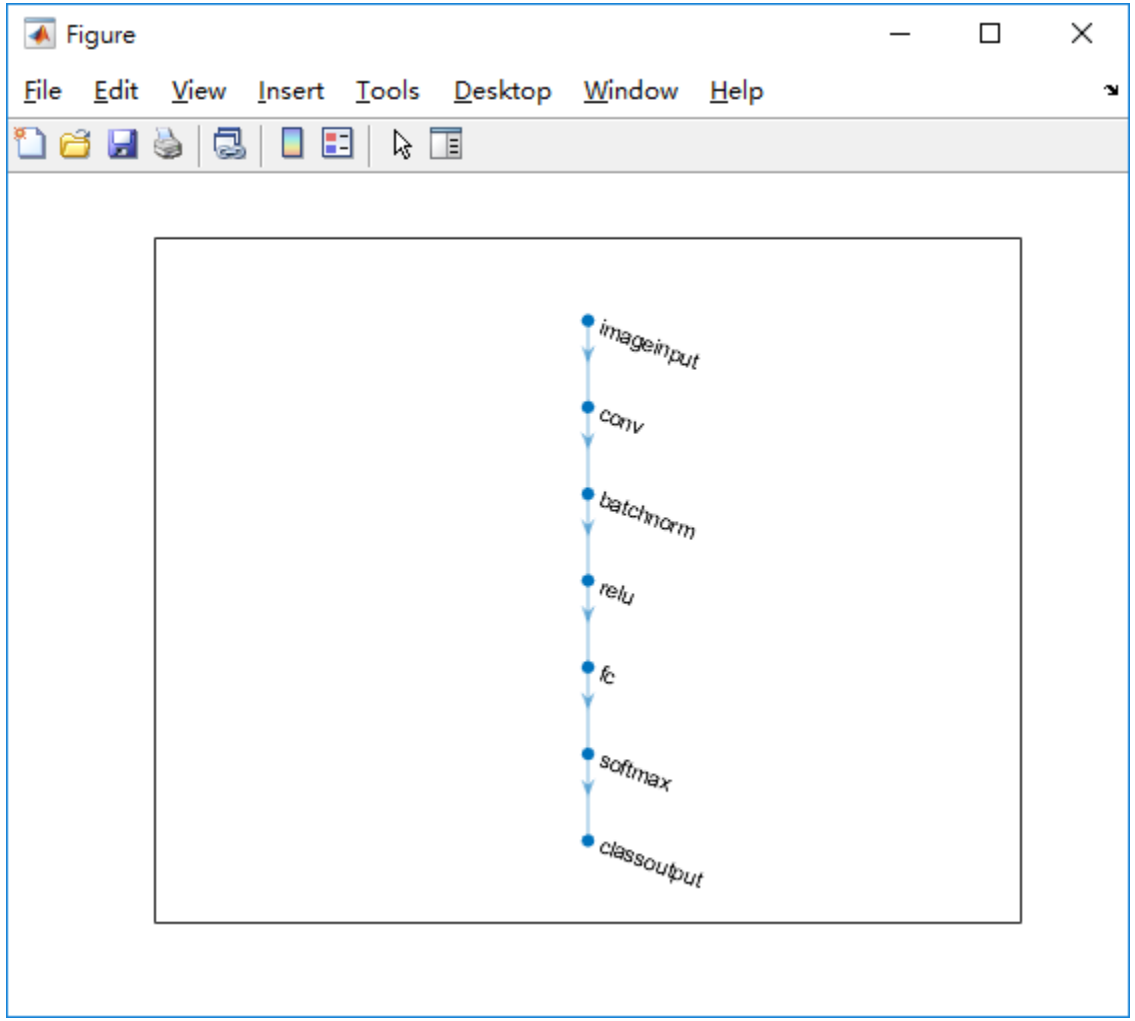
Layers can be ordered in different ways.



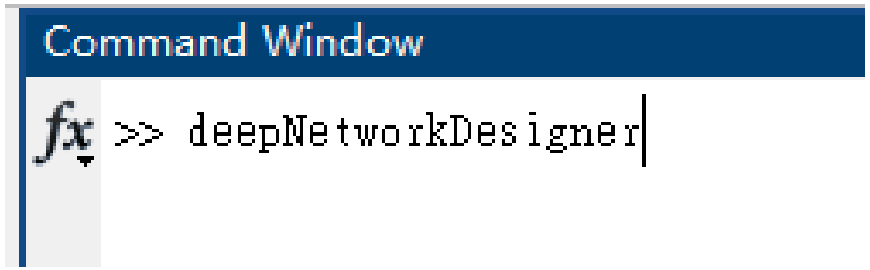
Building Convolution Neural Network

Create Array of Layers

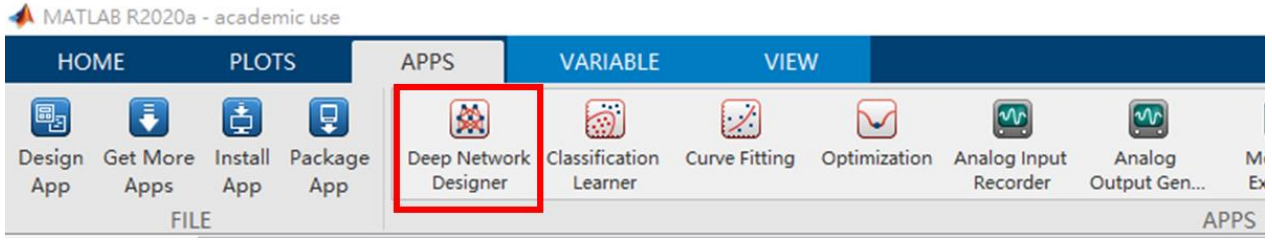
```
layers = [
imageInputLayer([28 28 1],"Name","imageinput")
convolution2dLayer([5 5],20,"Name","conv")
batchNormalizationLayer("Name","batchnorm")
reluLayer("Name","relu")
fullyConnectedLayer(10,"Name","fc")
softmaxLayer("Name","softmax")
classificationLayer("Name","classoutput")];

plot(layerGraph(layers));
```

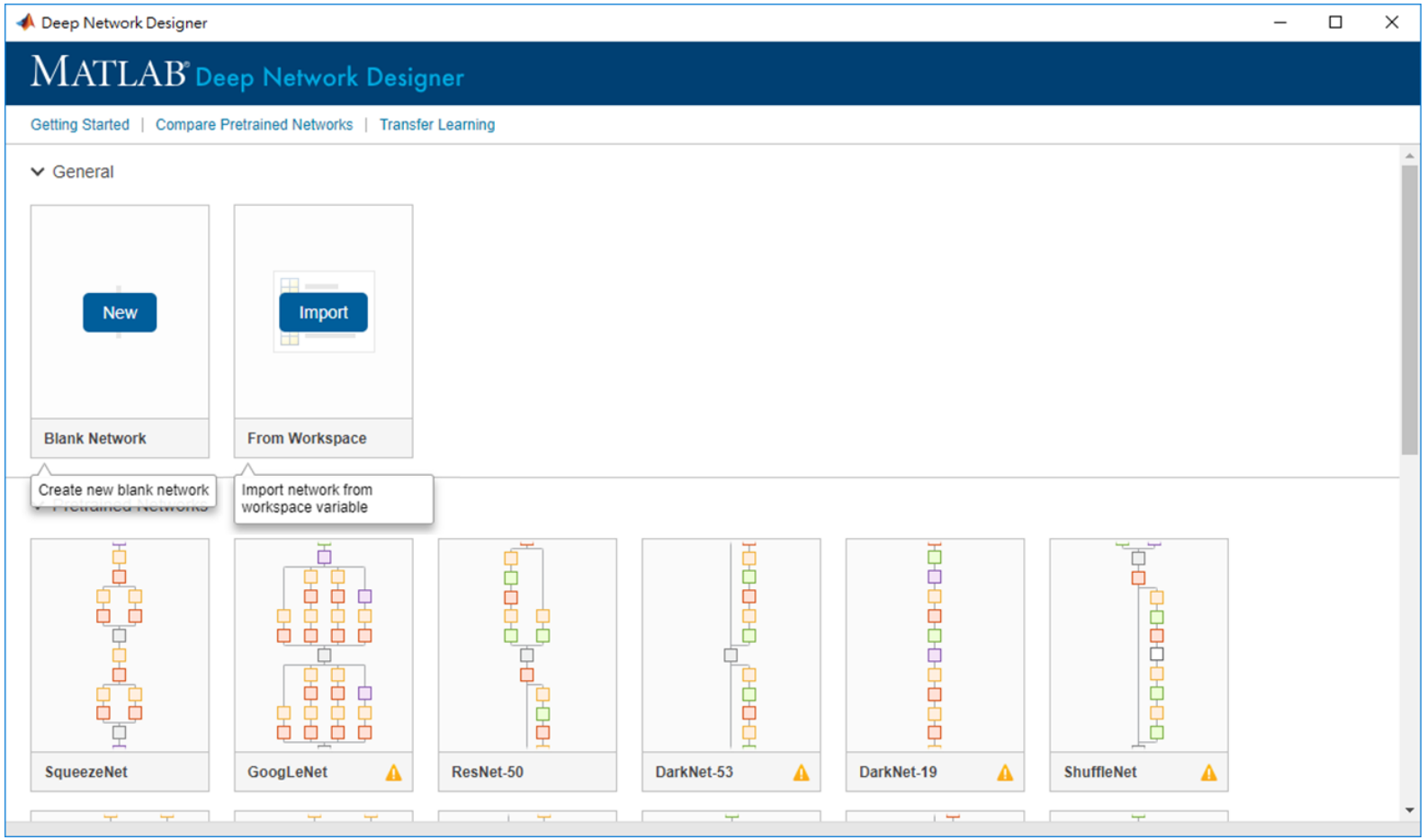


Deep Network Designer



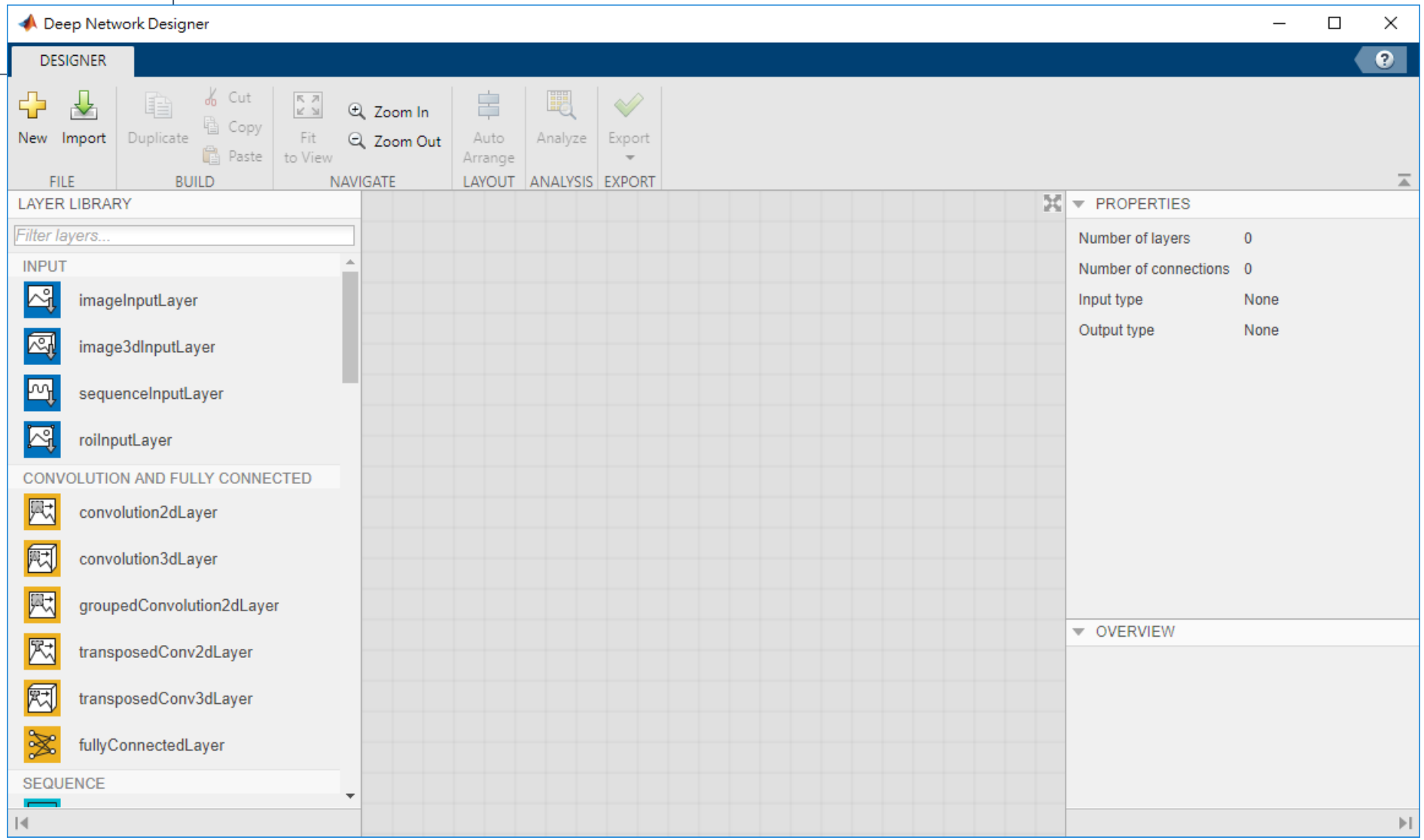
or





R2019b

What's New in Simulink



Deep Network Designer

INPUT
- imageInputLayer
- image3dInputLayer
- sequenceInputLayer
- roiInputLayer

CONVOLUTION AND FULLY CONNECTED
- convolution2dLayer
- convolution3dLayer
- groupedConvolution2dLayer
- transposedConv2dLayer
- transposedConv3dLayer
- fullyConnectedLayer

SEQUENCE
- lstmLayer
- bilstmLayer
- gruLayer
- sequenceFoldingLayer
- sequenceUnfoldingLayer
- flattenLayer
- wordEmbeddingLayer

OBJECT DETECTION
- regionProposalLayer
- yolov2ReorgLayer
- yolov2TransformLayer
- anchorBoxLayer
- ssdMergeLayer

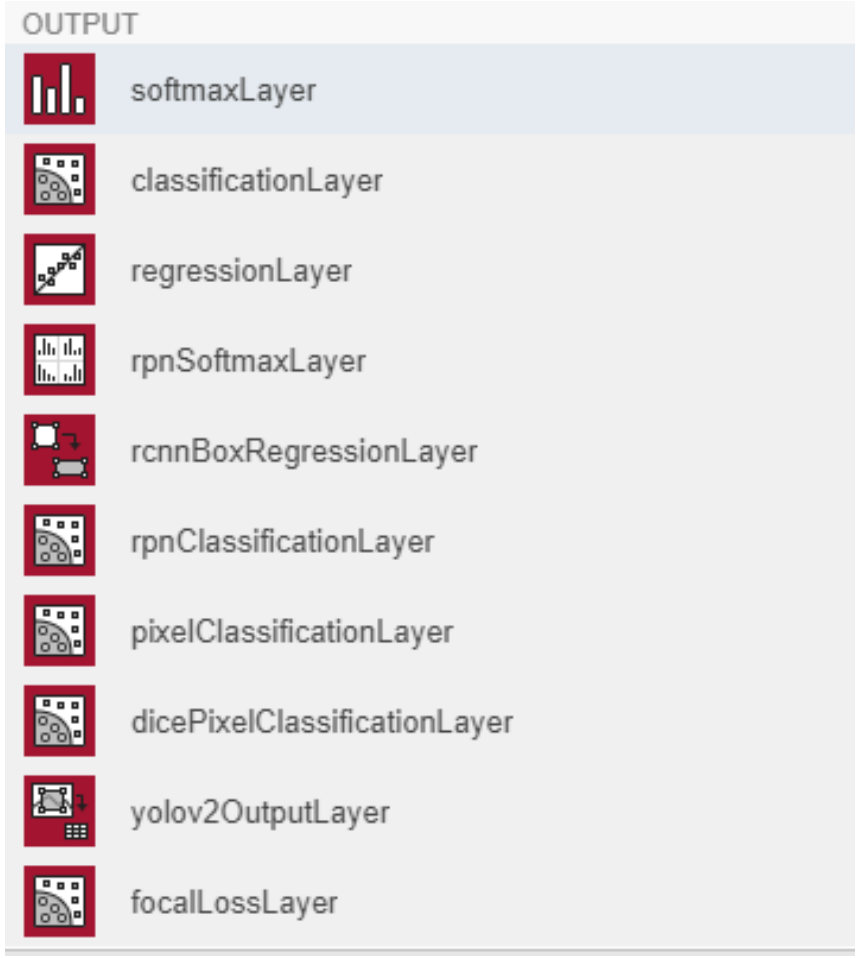
ACTIVATION
- reluLayer
- leakyReluLayer
- clippedReluLayer
- tanhLayer
- eluLayer
- softplusLayer

NORMALIZATION AND UTILITY
- dropoutLayer
- batchNormalizationLayer
- crossChannelNormalizationLayer
- crop2dLayer
- crop3dLayer
- scalingLayer
- quadraticLayer

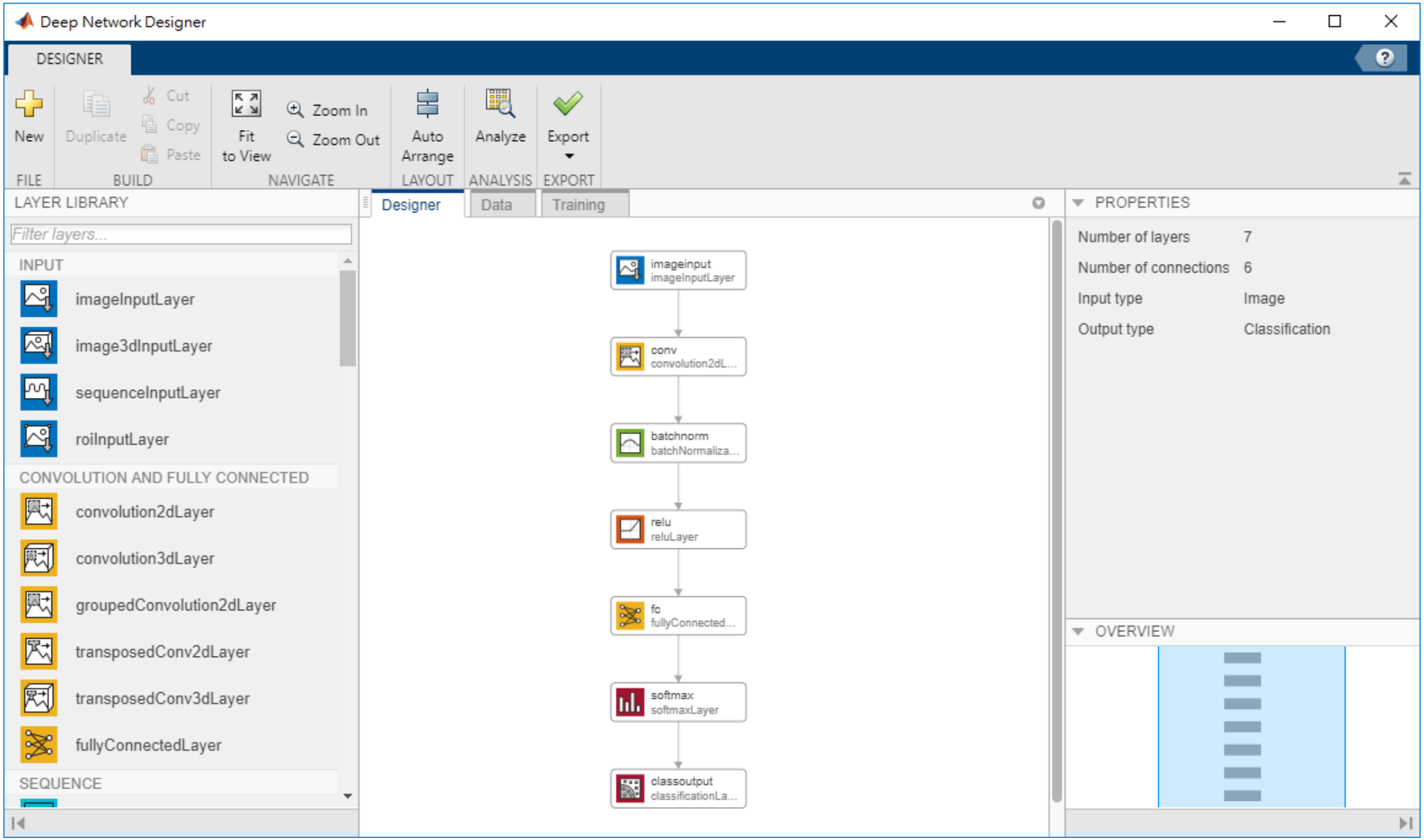
COMBINATION
- additionLayer
- depthConcatenationLayer
- concatenationLayer

POOLING
- averagePooling2dLayer
- averagePooling3dLayer
- globalAveragePooling2dLayer
- globalAveragePooling3dLayer
- maxPooling2dLayer
- maxPooling2dLayer (for unpooling)
- maxUnpooling2dLayer
- maxPooling3dLayer
- globalMaxPooling2dLayer
- globalMaxPooling3dLayer
- roiMaxPooling2dLayer

Deep Network Designer



Deep Network Designer

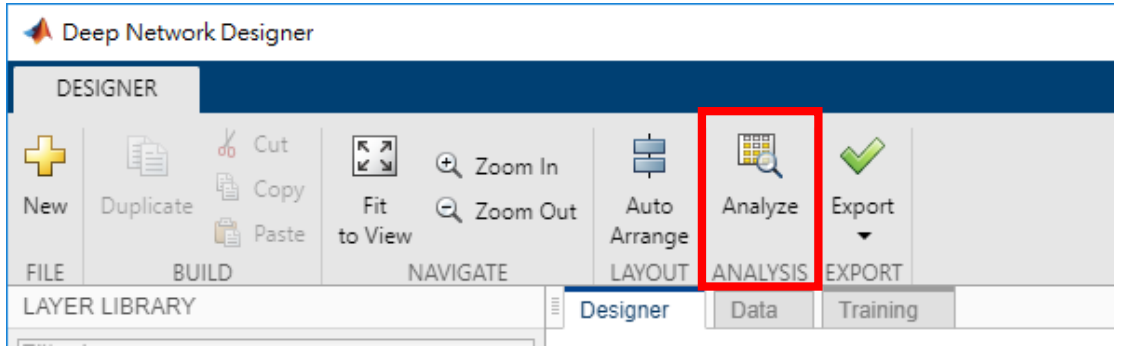


Deep Network Analyzer

Command Window

```
>> analyzeNetwork(layers)
```

or



Deep Learning Network Analyzer

Network from Deep Network Designer

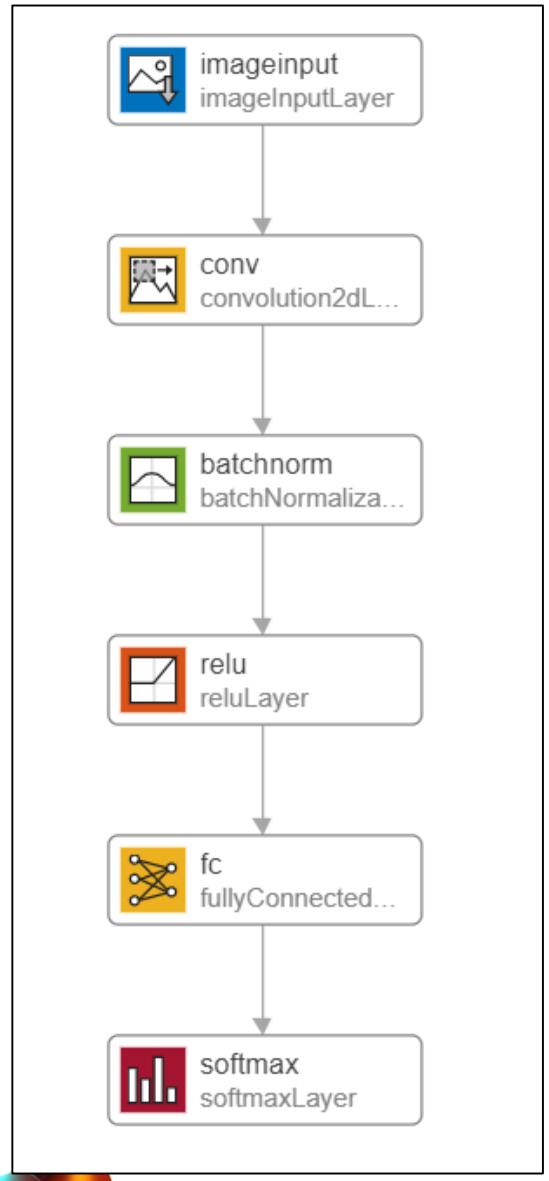
Analysis date: 27-May-2020 23:06:39

7 layers | 0 warnings | 0 errors

ANALYSIS RESULT

	Name	Type	Activations	Learnables
1	imageinput 28x28x1 images with 'zerocenter' normalization	Image Input	28×28×1	-
2	conv 20 5x5x1 convolutions with stride [1 1] and padding [0 0 0 0]	Convolution	24×24×20	Weights 5×5×1×20 Bias 1×1×20
3	batchnorm Batch normalization with 20 channels	Batch Normalization	24×24×20	Offset 1×1×20 Scale 1×1×20
4	relu ReLU	ReLU	24×24×20	-
5	fc 10 fully connected layer	Fully Connected	1×1×10	Weights 10×11520 Bias 10×1
6	softmax softmax	Softmax	1×1×10	-
7	classoutput crossentropyex	Classification Output	-	-

Deep Network Analyzer



Deep Learning Network Analyzer

Network from Deep Network Designer

Analysis date: 27-May-2020 23:14:01

6 layers | 0 warnings | 2 errors

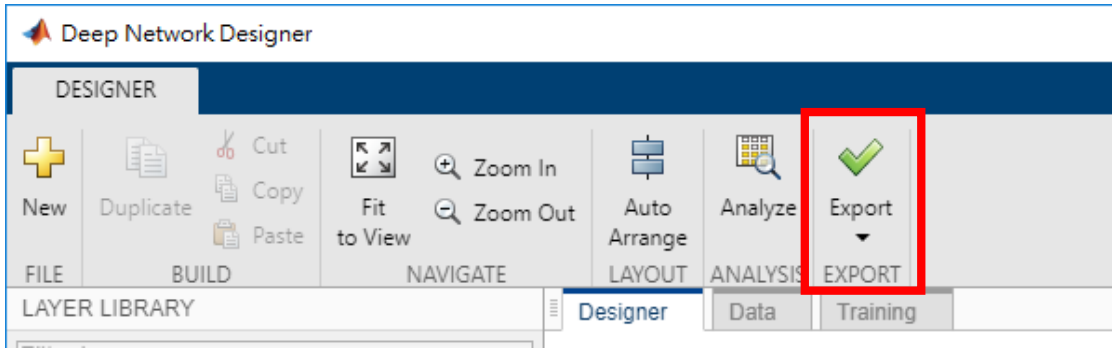
ISSUES

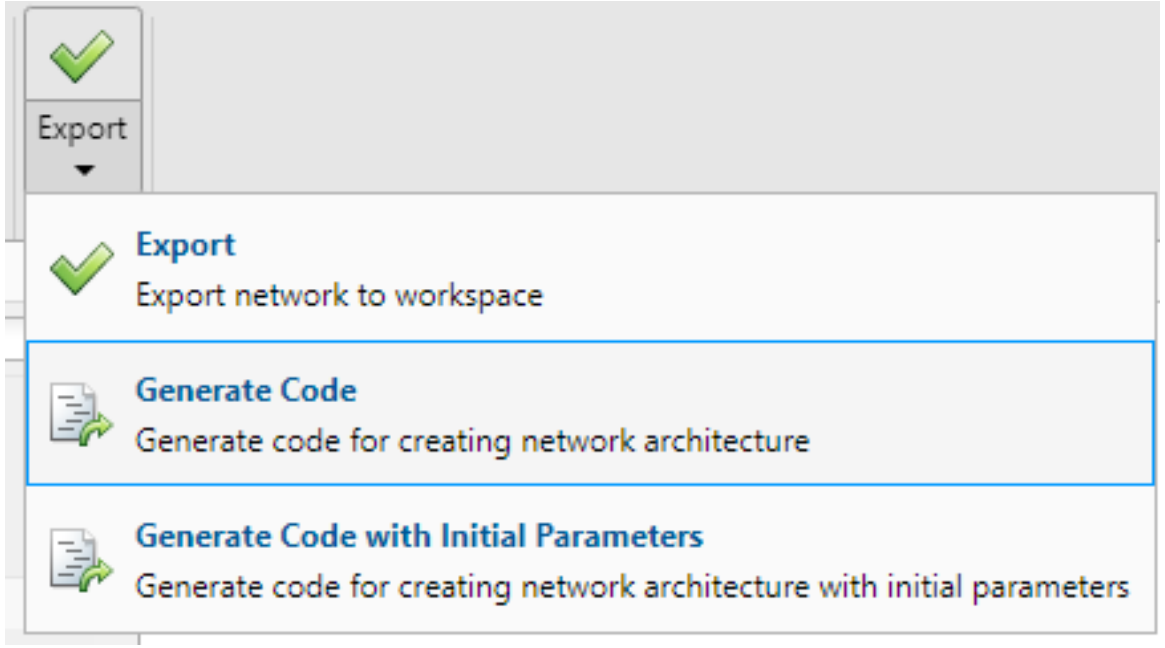
	Found in	Message
!	softmax	Unconnected output. Each layer output must be connected to the input of another layer.
!	Network	Missing output layer. The network must have at least one output layer.

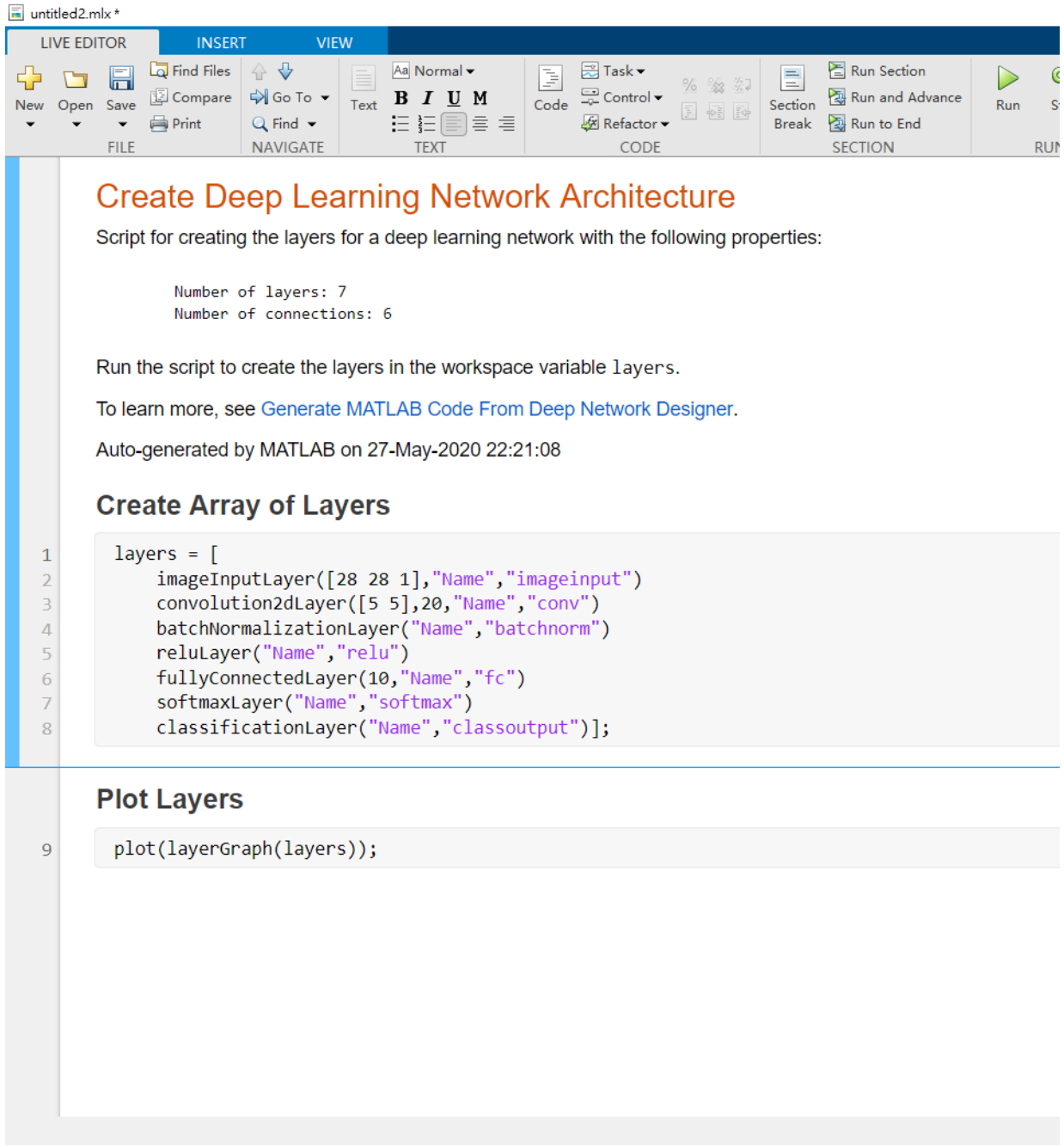
ANALYSIS RESULT

	Name	Type	Activations	Learnables
1	imageinput 28x28x1 images with 'zerocenter' normalization	Image Input	28×28×1	-
2	conv 20 5x5x1 convolutions with stride [1 1] and padding [0 0 0 0]	Convolution	24×24×20	Weights 5×5×1×20 Bias 1×1×20
3	batchnorm Batch normalization with 20 channels	Batch Normalization	24×24×20	Offset 1×1×20 Scale 1×1×20
4	relu ReLU	ReLU	24×24×20	-
5	fc 10 fully connected layer	Fully Connected	1×1×10	Weights 10×11520 Bias 10×1
6	! softmax softmax	Softmax	1×1×10	-

Export Model



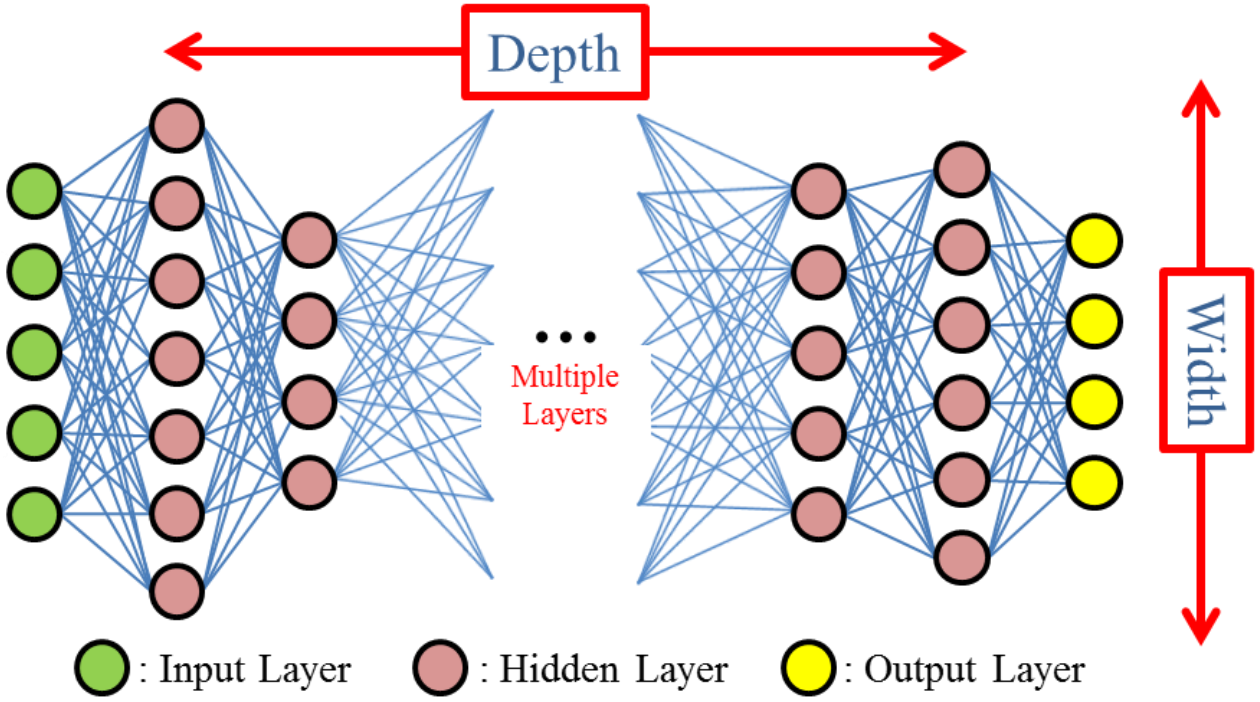




HyperParameters

In machine learning, a hyperparameter is a parameter whose value is set before the learning process begins.

By contrast, the values of other parameters are derived via training. (weighting, bias, …)



Optimizer
Initial Learn Rate
Learn Rate Drop Factor
Max Epochs
Mini-batch size
L2 Regularization
Gradient Decay Factor
Squared Gradient Decay Factor

⋮

HyperParameters

```
option = trainingOptions("adam")
```

```
trainingOptions(solverName,options)
```

Solver for training network

- "adam"
- "rmsprop"
- "sgdm"

```
option = trainingOptions("adam")
```

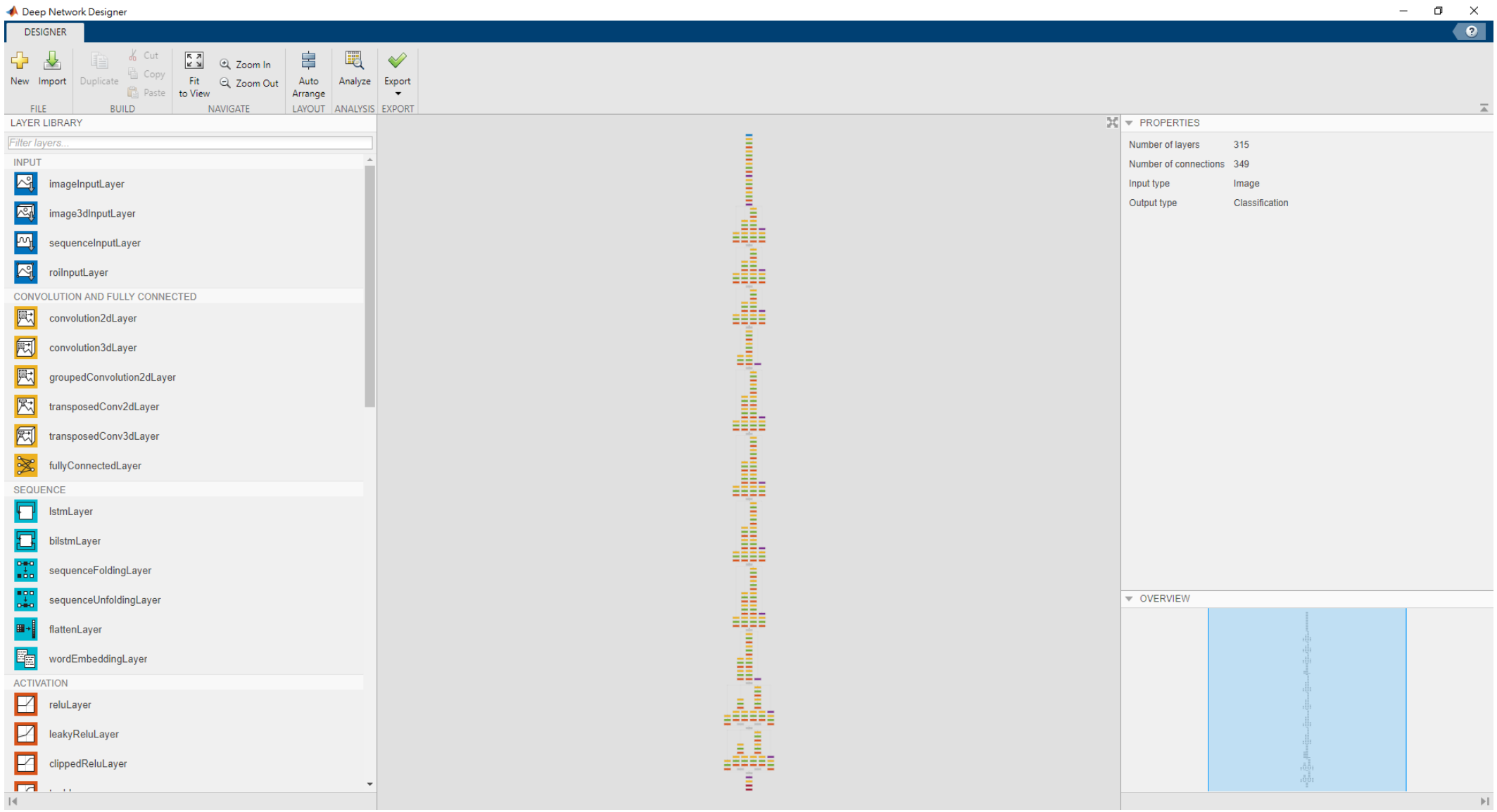
```
option =
  TrainingOptionsADAM with properties:

            GradientDecayFactor: 0.9000
        SquaredGradientDecayFactor: 0.9990
                        Epsilon: 1.0000e-08
                InitialLearnRate: 1.0000e-03
       LearnRateScheduleSettings: [1×1 struct]
                L2Regularization: 1.0000e-04
          GradientThresholdMethod: 'l2norm'
                GradientThreshold: Inf
                        MaxEpochs: 30
                    MiniBatchSize: 128
                          Verbose: 1
                  VerboseFrequency: 50
                    ValidationData: []
               ValidationFrequency: 50
                ValidationPatience: Inf
                          Shuffle: 'once'
                   CheckpointPath: ''
             ExecutionEnvironment: 'auto'
                       WorkerLoad: []
                        OutputFcn: []
                            Plots: 'none'
                   SequenceLength: 'longest'
              SequencePaddingValue: 0
           SequencePaddingDirection: 'right'
              DispatchInBackground: 0
           ResetInputNormalization: 1
```

Use Pretrained Model

Network	Depth	Size	Parameters (Millions)	Image Input Size
squeezenet	18	4.6 MB	1.24	227-by-227
googlenet	22	27 MB	7.0	224-by-224
inceptionv3	48	89 MB	23.9	299-by-299
densenet201	201	77 MB	20.0	224-by-224
mobilenetv2	53	13 MB	3.5	224-by-224
resnet18	18	44 MB	11.7	224-by-224
resnet50	50	96 MB	25.6	224-by-224
resnet101	101	167 MB	44.6	224-by-224
xception	71	85 MB	22.9	299-by-299
inceptionresnetv2	164	209 MB	55.9	299-by-299
shufflenet	50	6.3 MB	1.4	224-by-224
nasnetmobile	*	20 MB	5.3	224-by-224
nasnetlarge	*	360 MB	88.9	331-by-331
darknet19	19	72.5 MB	21.0	256-by-256
darknet53	53	145 MB	41.0	256-by-256
alexnet	8	227 MB	61.0	227-by-227
vgg16	16	515 MB	138	224-by-224
vgg19	19	535 MB	144	224-by-224

Use Pretrained Model

inceptionv3



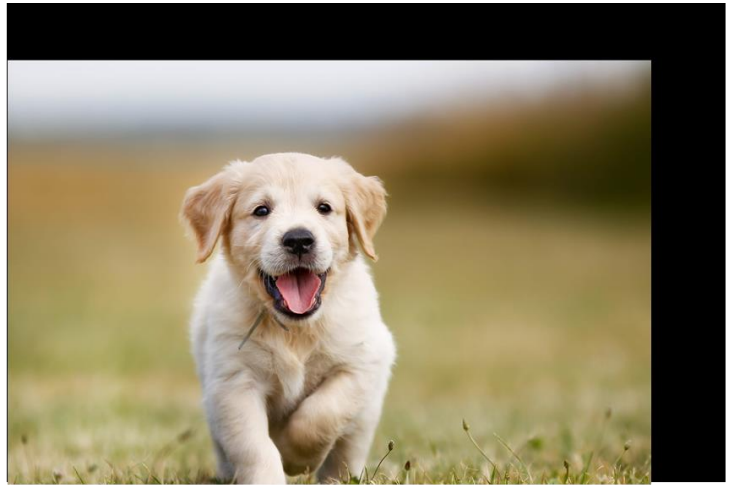
Layers、SeriesNetwork、DAG(directed acyclic graph)

- connectLayers
- addLayers
- removeLayers
- replaceLayer

Data Augmentation

Rescale

Translation



Rotation

Shear