NNV + GAN Intro

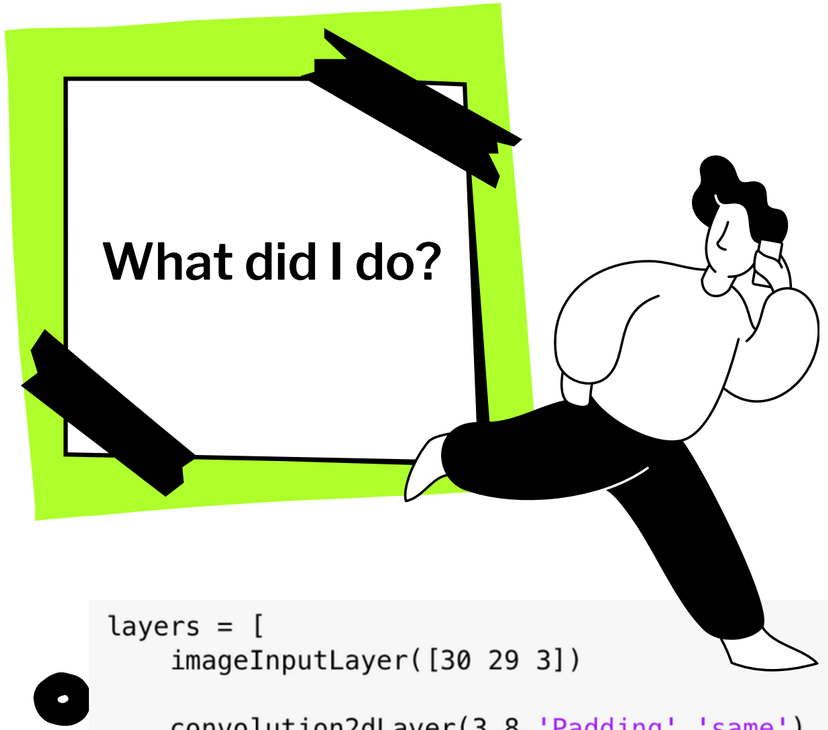
STUDENT: CHRISTINE ZHOU
PI: TAYLOR JOHNSON

NNV = Neural Network Verification

- *verify* the behavior learned by deep neural networks
- detect or prove the absence of perturbations that can cause various computer vision and machine perception tasks to *misbehave*

Goal for Project

- Develop *scripts* in Matlab for performing benchmarking of Professor Johnson's recent research
- *Verifying* these scripts with famous neural network databases
- Train a Generative Adversarial Network (GAN) in MATLAB that can generate data with similar characteristics as the input real data




```
layers = [
    imageInputLayer([30 29 3])

    convolution2dLayer(3,8,'Padding','same')
    batchNormalizationLayer
    reluLayer

    maxPooling2dLayer(2,'Stride',2)

    convolution2dLayer(3,16,'Padding','same')
    batchNormalizationLayer
    reluLayer

    maxPooling2dLayer(2,'Stride',2)

    convolution2dLayer(3,32,'Padding','same')
    batchNormalizationLayer
    reluLayer

    fullyConnectedLayer(43)
    softmaxLayer
    classificationLayer];
```

What I Did

- Learned the basics about neural network verification by finishing an online course
- Created a Deep Learning Network for Classification and tested it with GTSRB (German Traffic Sign Benchmarks) database (Final accuracy ~95.7%), snippet see left.
- Simple neural network: convolution layer + batch normalization + relu activation layer (increasing popularity, reduces overfitting)+pooling
- Trained a generative adversarial network (GAN) to generate images in MATLAB

```
layersGenerator = [
    imageInputLayer([1 1 numLatentInputs],'Normalization','none','Name','in')
    projectAndReshapeLayer(projectionSize,numLatentInputs,'proj');
    transposedConv2dLayer(filterSize,4*numFilters,'Name','tconv1')
    batchNormalizationLayer('Name','bnorm1')
    reluLayer('Name','relu1')
    transposedConv2dLayer(filterSize,2*numFilters,'Stride',2,'Cropping','same','Name','tconv2')
    batchNormalizationLayer('Name','bnorm2')
    reluLayer('Name','relu2')
    transposedConv2dLayer(filterSize,numFilters,'Stride',2,'Cropping','same','Name','tconv3')
    batchNormalizationLayer('Name','bnorm3')
    reluLayer('Name','relu3')
    transposedConv2dLayer(filterSize,3,'Stride',2,'Cropping','same','Name','tconv4')
    tanhLayer('Name','tanh')];

lgraphGenerator = layerGraph(layersGenerator);

dlnetGenerator = dlnetwork(lgraphGenerator);

dropoutProb = 0.5;
numFilters = 64;
scale = 0.2;

inputSize = [64 64 3];
```

```
layersDiscriminator = [
    imageInputLayer(inputSize,'Normalization','none','Name','in')
    dropoutLayer(0.5,'Name','dropout')
    convolution2dLayer(filterSize,numFilters,'Stride',2,'Padding','same','Name','conv1')
    leakyReluLayer(scale,'Name','lrelu1')
    convolution2dLayer(filterSize,2*numFilters,'Stride',2,'Padding','same','Name','conv2')
    batchNormalizationLayer('Name','bn2')
    leakyReluLayer(scale,'Name','lrelu2')
    convolution2dLayer(filterSize,4*numFilters,'Stride',2,'Padding','same','Name','conv3')
    batchNormalizationLayer('Name','bn3')
    leakyReluLayer(scale,'Name','lrelu3')
    convolution2dLayer(filterSize,8*numFilters,'Stride',2,'Padding','same','Name','conv4')
    batchNormalizationLayer('Name','bn4')
    leakyReluLayer(scale,'Name','lrelu4')
    convolution2dLayer(4,1,'Name','conv5')];

lgraphDiscriminator = layerGraph(layersDiscriminator);

dlnetDiscriminator = dlnetwork(lgraphDiscriminator);
```

Snippets of My Code for my GAN

- The generator learns to generate plausible data. The generated instances become negative training examples for the discriminator.
- The discriminator learns to distinguish the generator's fake data from real data. It takes an image as input, passes through convolution stacks and output a probability (sigmoid value). The discriminator penalizes the generator for producing implausible results (0.5 as threshold).
- Backpropagation: adjusts each weight and bias by calculating the weight's impact on the output
- Competing (thus "adversarial"). The discrimantor trains the generator.

My GAN

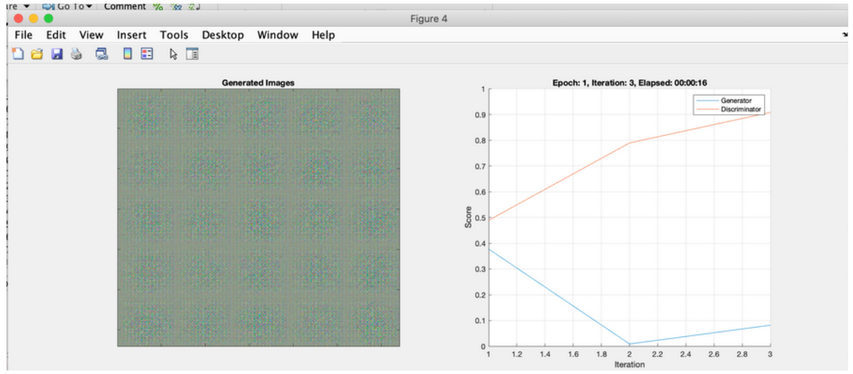
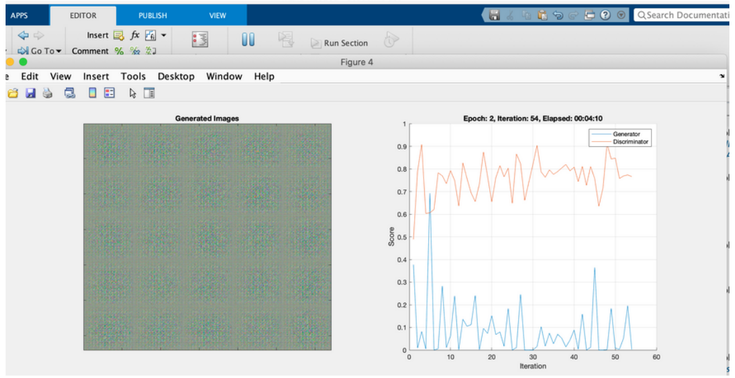
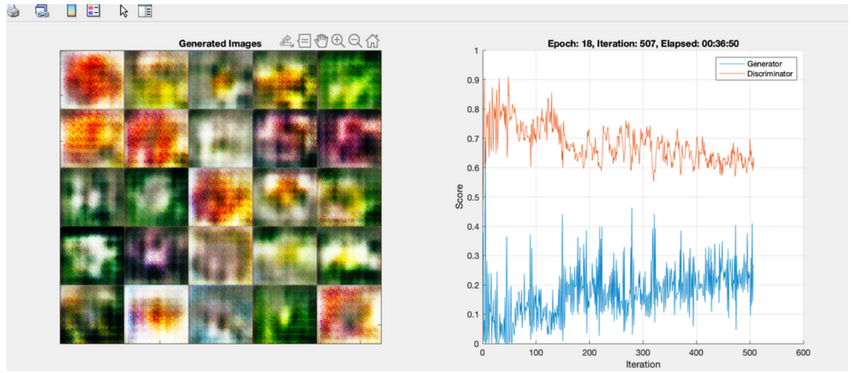
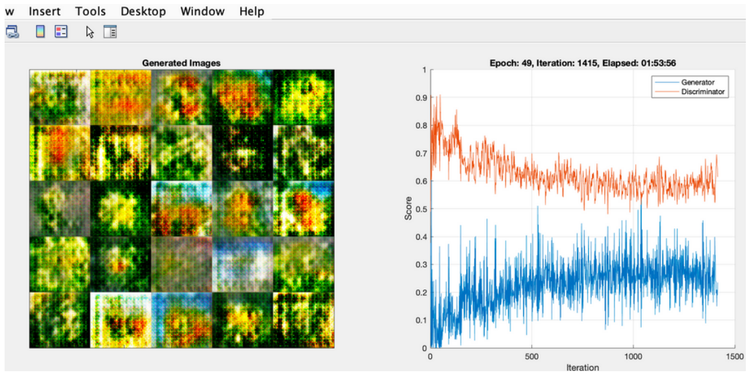
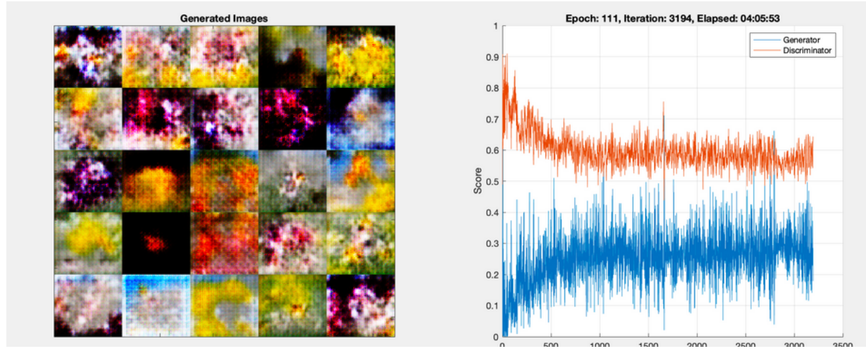


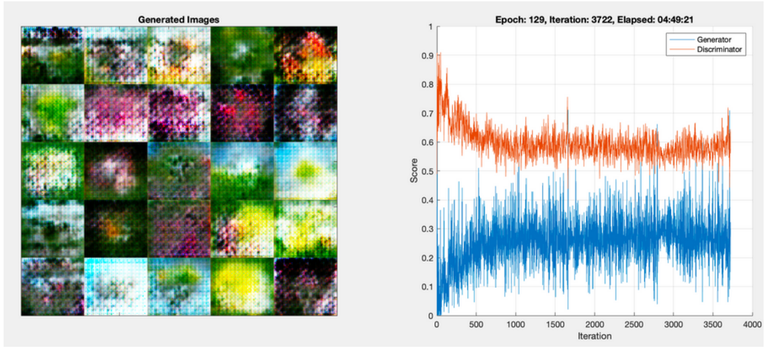
Figure 4
File Edit View Insert Tools Desktop Window Help
Generated Images
Epoch: 1, Iteration: 3, Elapsed: 00:00:16
Generator
Discriminator
Score
Iteration


APPS
EDITOR
PUBLISH
VIEW
Figure 4
Edit View Insert Tools Desktop Window Help
Generated Images
Epoch: 2, Iteration: 54, Elapsed: 00:04:10
Generator
Discriminator
Score
Iteration


Generated Images
Epoch: 18, Iteration: 507, Elapsed: 00:36:50
Generator
Discriminator
Score
Iteration


w Insert Tools Desktop Window Help
Generated Images
Epoch: 49, Iteration: 1415, Elapsed: 01:53:56
Generator
Discriminator
Score
Iteration


Generated Images
Epoch: 111, Iteration: 3194, Elapsed: 04:05:53
Generator
Discriminator
Score


Generated Images
Epoch: 129, Iteration: 3722, Elapsed: 04:49:21
Generator
Discriminator
Score
Iteration