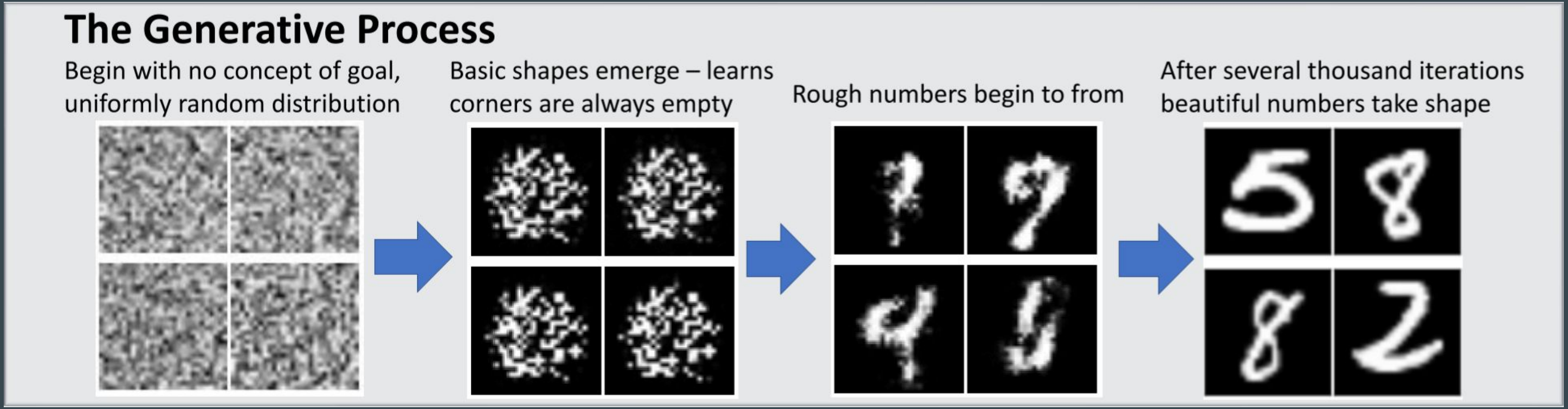
InfoGAN : New Type of Facial Recognition

Kastan Day & Tim Nguyen

OpenAI

Refresher

- InfoGAN
- Coevolution
- Generates useful images



The Generative Process

Begin with no concept of goal, uniformly random distribution

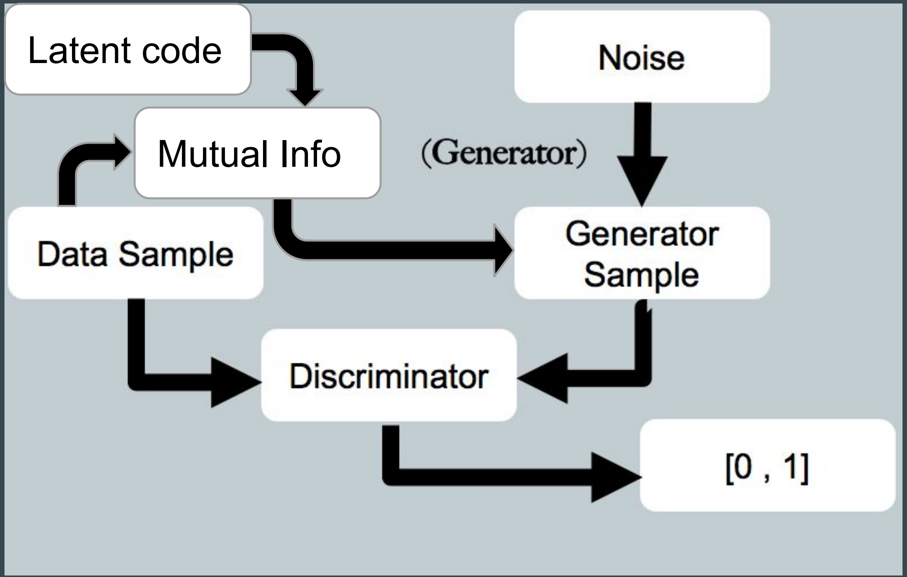
Basic shapes emerge – learns corners are always empty

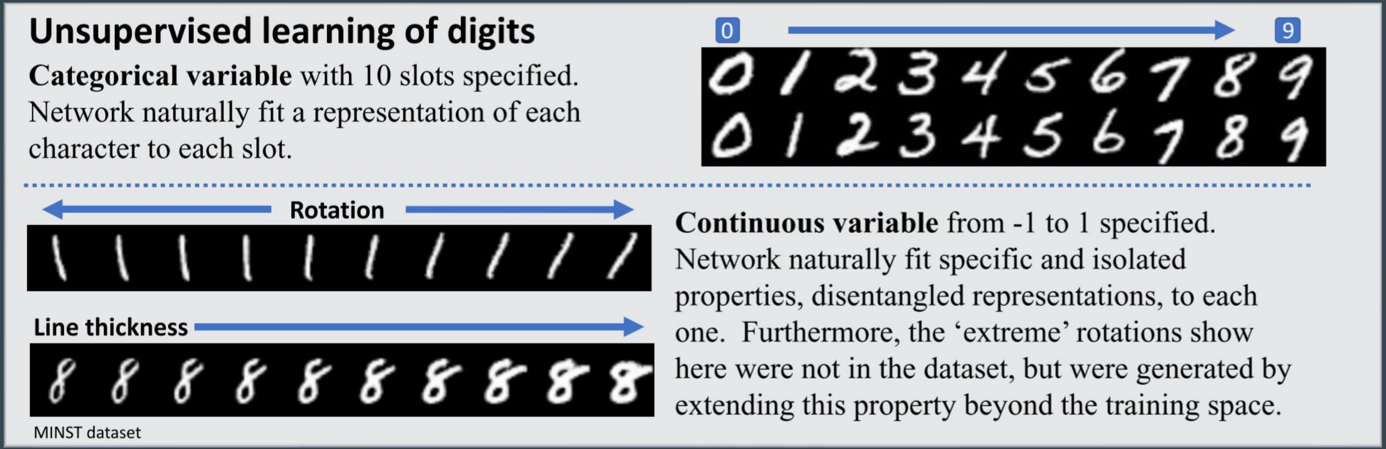
Rough numbers begin to from

After several thousand iterations beautiful numbers take shape

Latent Codes

- Specify variables
- Let InfoGAN fit data to variables





Unsupervised learning of digits

Categorical variable with 10 slots specified. Network naturally fit a representation of each character to each slot.

0
9
Rotation
Line thickness

Continuous variable from -1 to 1 specified. Network naturally fit specific and isolated properties, disentangled representations, to each one. Furthermore, the ‘extreme’ rotations show here were not in the dataset, but were generated by extending this property beyond the training space.

MINST dataset

Loss Functions

$$I(c, G(z,c)) = \mathbb{E}_{c \sim P(c), x \sim G(z,c)} [\log Q(c|X)] + H(c)$$

$$loss = \min_{G} \max_{D} \; V(D,G) - \; \lambda \, I(c, G(z,c))$$

Loss Functions

- Information Maximization

$$I(c, G(z,c)) = \mathbb{E}_{c\sim P(c), x\sim G(z,c)} [\log Q(c|X)] + H(c)$$

$$loss = \underset{}{min\,max} \quad (D, G) - \lambda\, I(c, G(z,c))$$

Later

Loss Functions

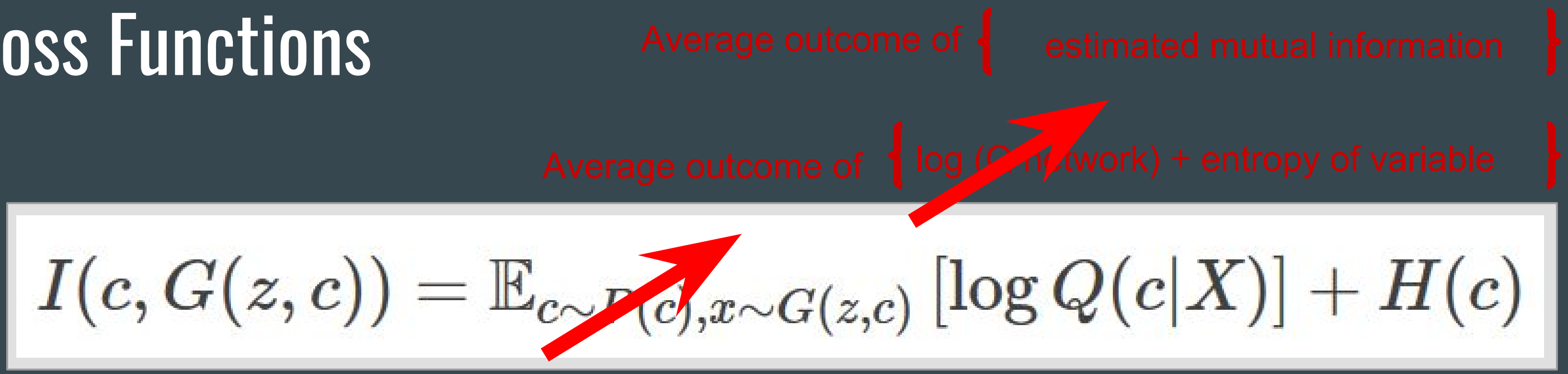
Average outcome of { log (Q network) + entropy of variable }

$$I(c, G(z,c)) = \mathbb{E}_{c \sim P(c), x \sim G(z,c)} [\log Q(c|X)] + H(c)$$

$$loss = \min \max (D, G) - \lambda\, I(c, G(z,c))$$

Later

Loss Functions



Average outcome of { estimated mutual information }

Average outcome of { log (Q network) + entropy of variable }

$$I(c, G(z,c)) = \mathbb{E}_{c\sim P(c), x\sim G(z,c)}[\log Q(c|X)] + H(c)$$

$loss = \ min\,max$ … $(D, G) - \lambda\, I(c, G(z,c))$

Later

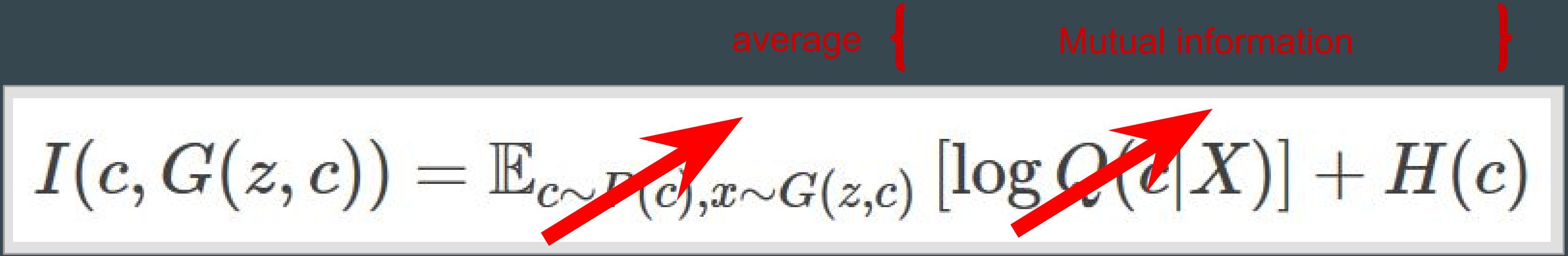
Loss Functions

$$I(c, G(z,c)) = \overset{\text{average}}{\mathbb{E}_{c\sim P(c), x\sim G(z,c)}} \overbrace{[\log Q(c|X)] + H(c)}^{\text{Mutual information}}$$

$$loss = min\,ma \quad (D, G) - \lambda\, I(c, G(z,c))$$

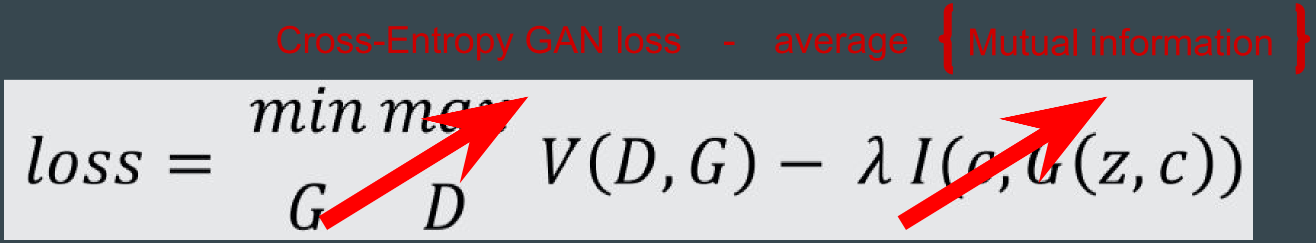
Later

Loss Functions



average { Mutual information }

$$I(c, G(z,c)) = \mathbb{E}_{c\sim P(c), x\sim G(z,c)}[\log Q(c|X)] + H(c)$$



Cross-Entropy GAN loss - average { Mutual information }

$$loss = \min_{G}\max_{D} V(D,G) - \lambda I(c, G(z,c))$$

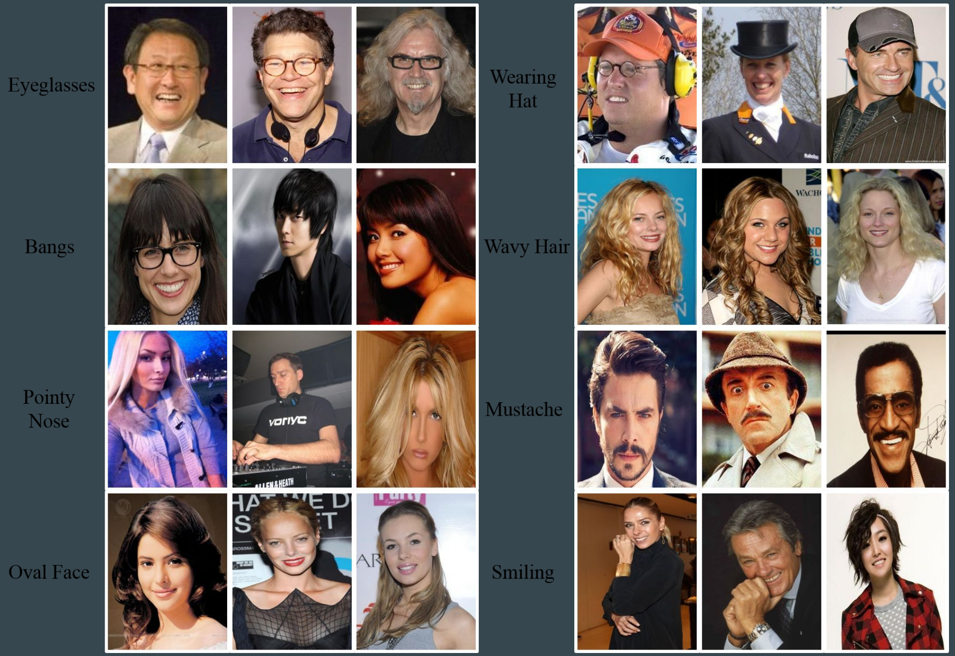
Goal

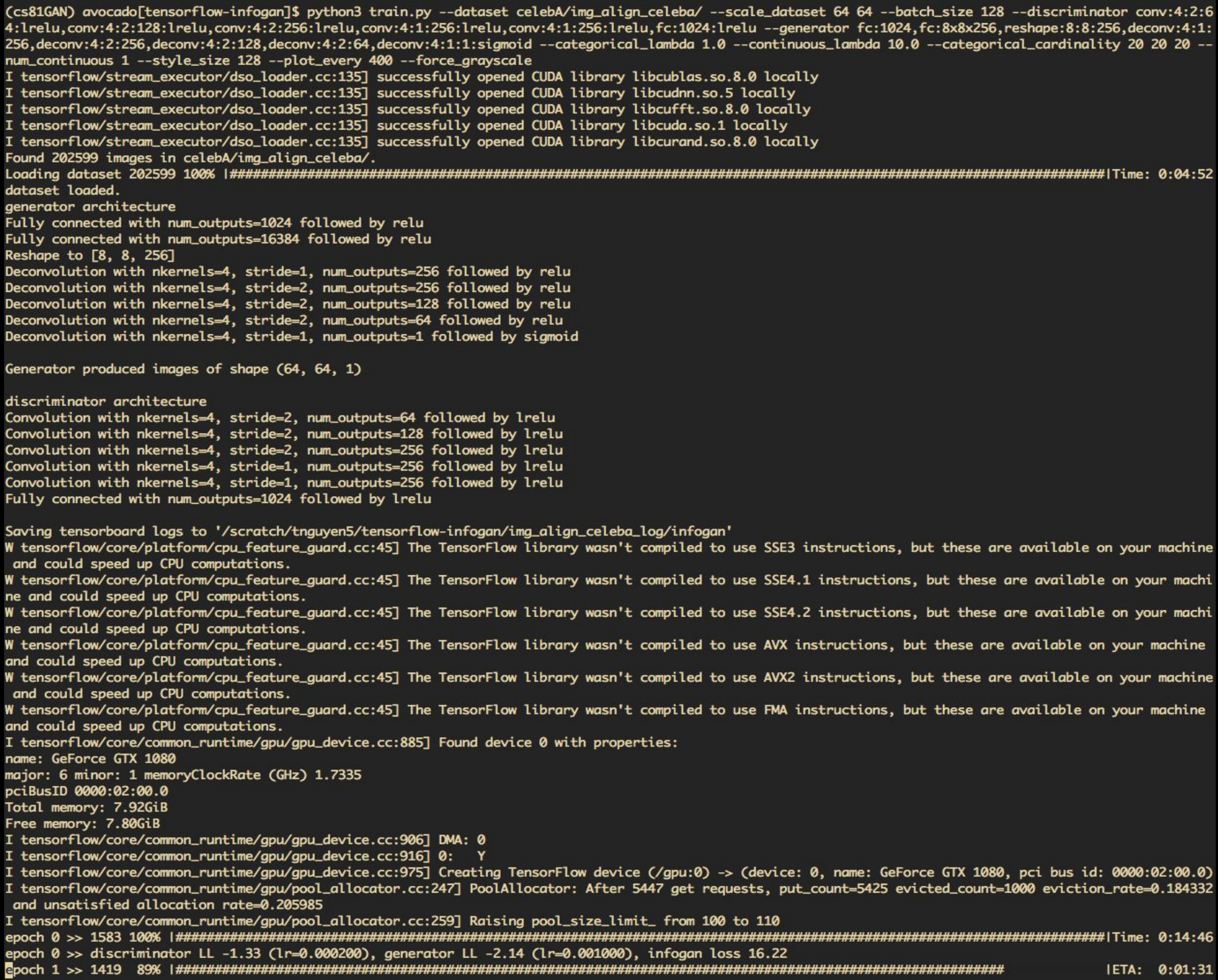
- Hypothesis : InfoGAN will learn meaningful, disentangled representations in an unsupervised, or semi-supervised way.
    - Facial Recognition
    - Animal recognition
    - Artistic style

Training

1) CelebA

2) George Bush

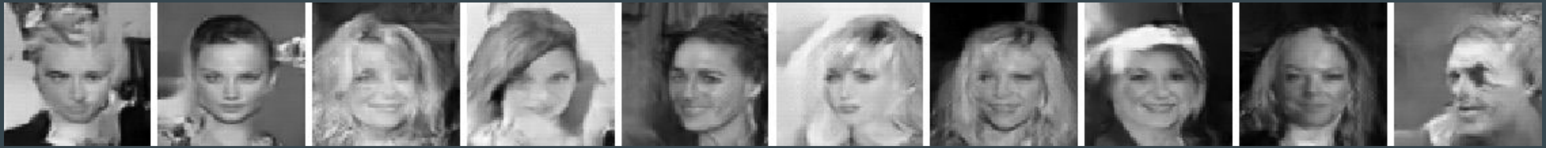




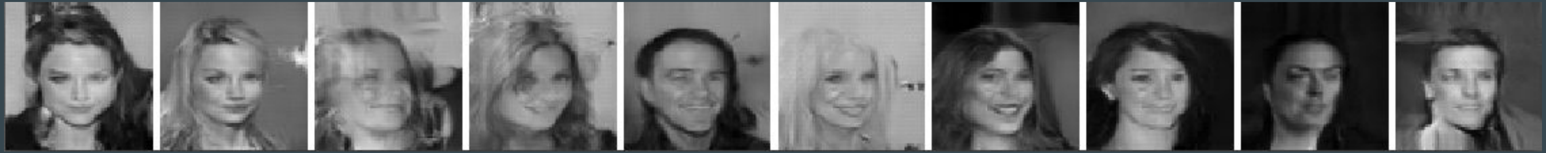
```
(cs81GAN) avocado[tensorflow-infogan]$ python3 train.py --dataset celebA/img_align_celeba/ --scale_dataset 64 64 --batch_size 128 --discriminator conv:4:2:6
4:lrelu,conv:4:2:128:lrelu,conv:4:2:256:lrelu,conv:4:1:256:lrelu,conv:4:1:256:lrelu,fc:1024:lrelu --generator fc:1024,fc:8x8x256,reshape:8:8:256,deconv:4:1:
256,deconv:4:2:256,deconv:4:2:128,deconv:4:2:64,deconv:4:1:1:sigmoid --categorical_lambda 1.0 --continuous_lambda 10.0 --categorical_cardinality 20 20 20 --
num_continuous 1 --style_size 128 --plot_every 400 --force_grayscale
I tensorflow/stream_executor/dso_loader.cc:135] successfully opened CUDA library libcublas.so.8.0 locally
I tensorflow/stream_executor/dso_loader.cc:135] successfully opened CUDA library libcudnn.so.5 locally
I tensorflow/stream_executor/dso_loader.cc:135] successfully opened CUDA library libcufft.so.8.0 locally
I tensorflow/stream_executor/dso_loader.cc:135] successfully opened CUDA library libcuda.so.1 locally
I tensorflow/stream_executor/dso_loader.cc:135] successfully opened CUDA library libcurand.so.8.0 locally
Found 202599 images in celebA/img_align_celeba/.
Loading dataset 202599 100% |#################################################################################################################|Time: 0:04:52
dataset loaded.
generator architecture
Fully connected with num_outputs=1024 followed by relu
Fully connected with num_outputs=16384 followed by relu
Reshape to [8, 8, 256]
Deconvolution with nkernels=4, stride=1, num_outputs=256 followed by relu
Deconvolution with nkernels=4, stride=2, num_outputs=256 followed by relu
Deconvolution with nkernels=4, stride=2, num_outputs=128 followed by relu
Deconvolution with nkernels=4, stride=2, num_outputs=64 followed by relu
Deconvolution with nkernels=4, stride=1, num_outputs=1 followed by sigmoid

Generator produced images of shape (64, 64, 1)

discriminator architecture
Convolution with nkernels=4, stride=2, num_outputs=64 followed by lrelu
Convolution with nkernels=4, stride=2, num_outputs=128 followed by lrelu
Convolution with nkernels=4, stride=2, num_outputs=256 followed by lrelu
Convolution with nkernels=4, stride=1, num_outputs=256 followed by lrelu
Convolution with nkernels=4, stride=1, num_outputs=256 followed by lrelu
Fully connected with num_outputs=1024 followed by lrelu

Saving tensorboard logs to '/scratch/tnguyen5/tensorflow-infogan/img_align_celeba_log/infogan'
W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use SSE3 instructions, but these are available on your machine
 and could speed up CPU computations.
W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use SSE4.1 instructions, but these are available on your machi
ne and could speed up CPU computations.
W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use SSE4.2 instructions, but these are available on your machi
ne and could speed up CPU computations.
W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use AVX instructions, but these are available on your machine
and could speed up CPU computations.
W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use AVX2 instructions, but these are available on your machine
 and could speed up CPU computations.
W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use FMA instructions, but these are available on your machine
and could speed up CPU computations.
I tensorflow/core/common_runtime/gpu/gpu_device.cc:885] Found device 0 with properties:
name: GeForce GTX 1080
major: 6 minor: 1 memoryClockRate (GHz) 1.7335
pciBusID 0000:02:00.0
Total memory: 7.92GiB
Free memory: 7.80GiB
I tensorflow/core/common_runtime/gpu/gpu_device.cc:906] DMA: 0
I tensorflow/core/common_runtime/gpu/gpu_device.cc:916] 0:   Y
I tensorflow/core/common_runtime/gpu/gpu_device.cc:975] Creating TensorFlow device (/gpu:0) -> (device: 0, name: GeForce GTX 1080, pci bus id: 0000:02:00.0)
I tensorflow/core/common_runtime/gpu/pool_allocator.cc:247] PoolAllocator: After 5447 get requests, put_count=5425 evicted_count=1000 eviction_rate=0.184332
 and unsatisfied allocation rate=0.205985
I tensorflow/core/common_runtime/gpu/pool_allocator.cc:259] Raising pool_size_limit_ from 100 to 110
epoch 0 >> 1583 100% |########################################################################################################################|Time: 0:14:46
epoch 0 >> discriminator LL -1.33 (lr=0.000200), generator LL -2.14 (lr=0.001000), infogan loss 16.22
epoch 1 >> 1419  89% |###########################################################################################################             |ETA:  0:01:31
```

CelebA (100 Epochs, 200k Images)

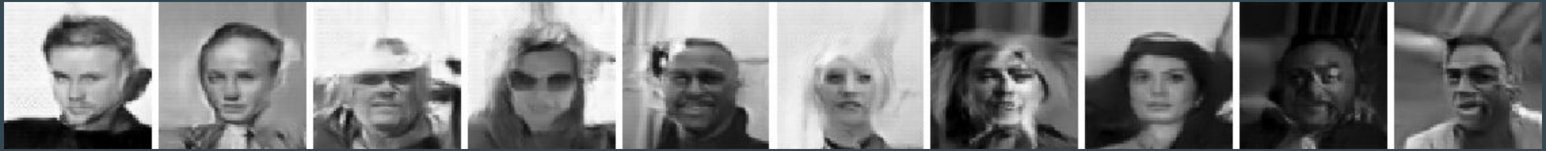
Categorical



Caucasian Women

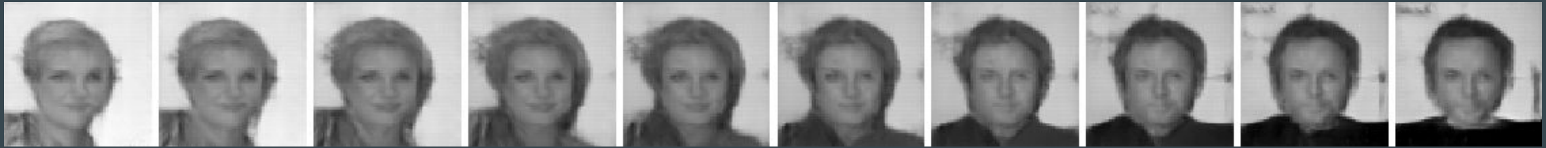


Facial Expression



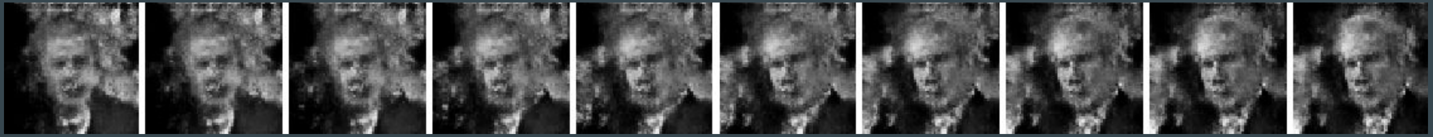
Darkening Skin

Continuous

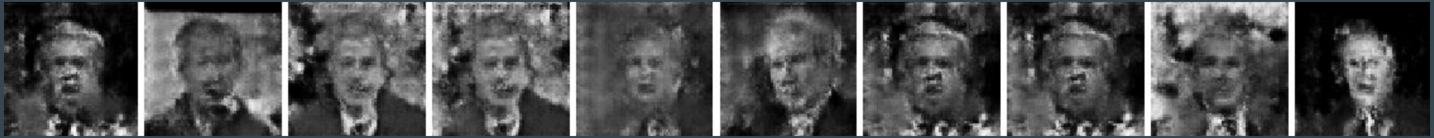


Continuous Variable

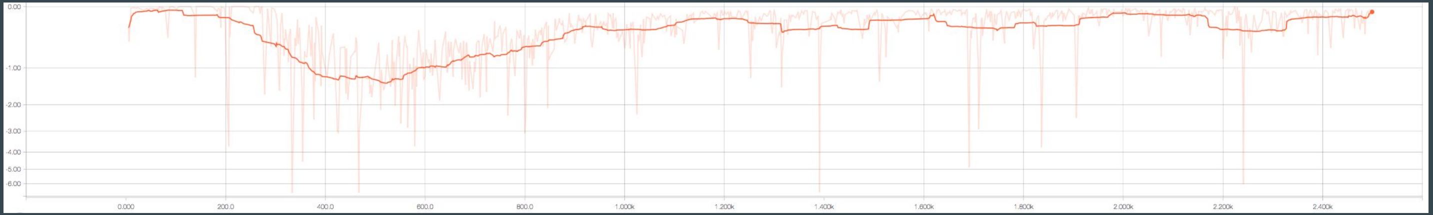
Bush (500 epochs, 560 Images)



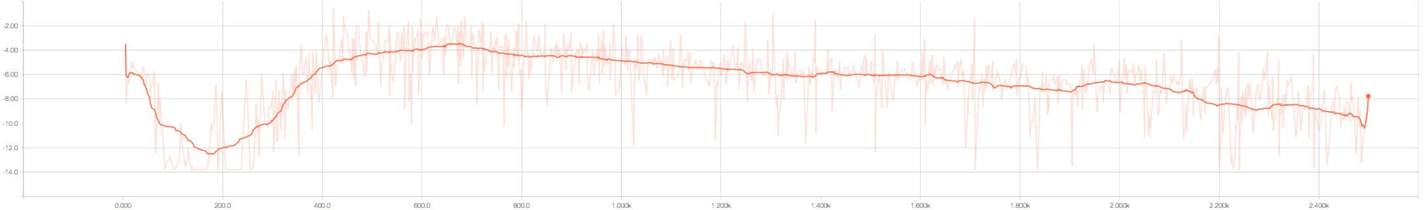
Continuous
Rotating Bush



Categorical
Speaking Bush

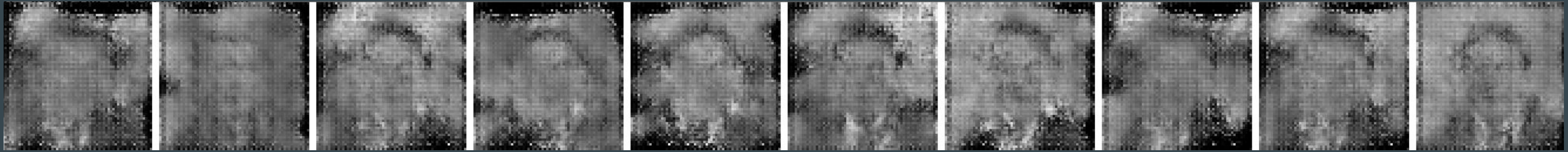


Discriminator
Objective

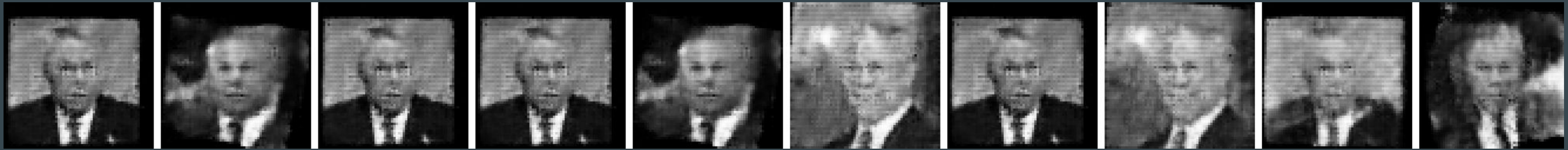


Generator
Objective

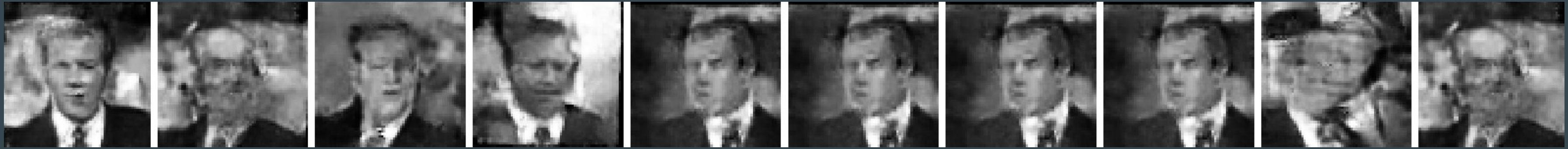
- YaleB11 (10,000 epochs, In Progress)
- Gerhard Schroeder (5000 epochs, Failed)



- Colin Powell (20,000 epochs, Stern Faced)



- Five People (Bush, Schroeder, Chavez, Blair, Sharon - Attempts at similarity)



Future Experiments

What we learned : GANs are hard! Tuning parameters are a pain.

- More data, longer training!
- Experiment with parameters.
- Incorporate many more different facial datasets.
- Determine correct balance between categorical and continuous variables.

Tensor Board Demo

(If we have time)