Applied Deep Learning

Day 4: GANs and Project Day

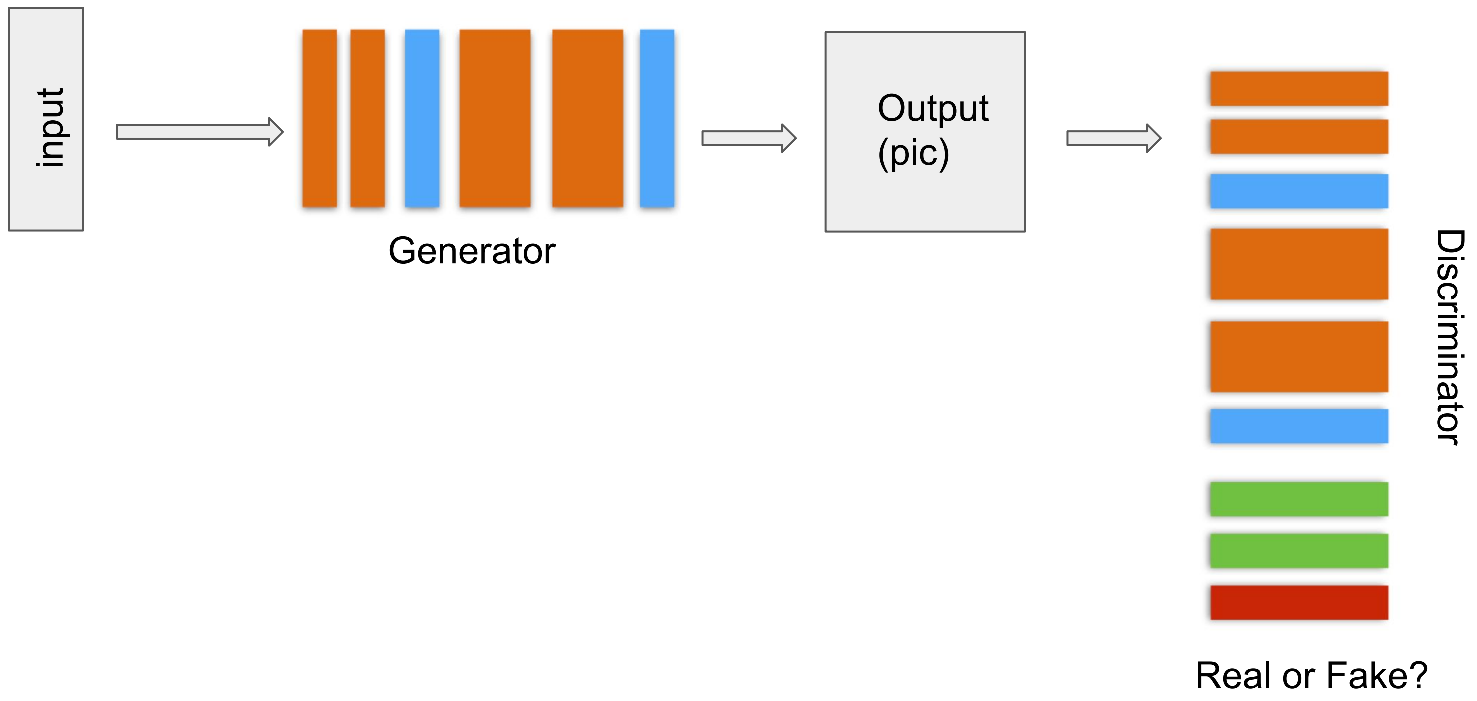
Generative Adversarial Networks

- Based on a game theory scenario in which a generator network must compete against a discriminator network
- The generator network is incentivized to create realistic looking examples of some data
- The discriminator network is incentivized to learn to distinguish fake examples from real examples

Generative Adversarial Networks

- Requires a dataset of real examples that the generator will try to mimic and the discriminator will try to learn
- Generator network will take some input, either a random vector or some structured input to use as a seed

Generative Adversarial Networks



Training Steps

- Multi step training
  - Train discriminator to distinguish real from fake
  - Train generator to make better fakes

Discriminator Training

- Send real and generated data through network
- Label data with appropriate classes, 0-fake 1-real
- Calculate loss for the two class problem
- Update discriminator weights according to this loss
    - Do not update generator weights at this step

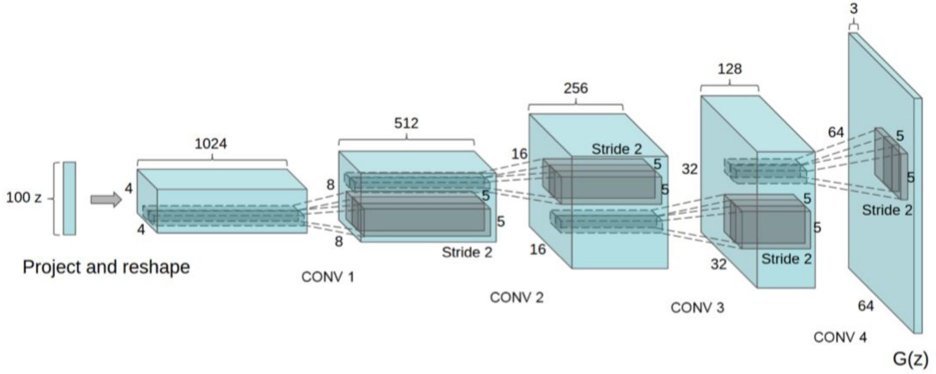
Generator Training

- Send the generated data through the discriminator and collect it’s predictions (can use the same predictions made during D training phase)
- This time set up the labels as if all images are of the ‘real’ class
- Calculate the loss with these labels
    - Update Generator weights accordingly

GAN Examples

- Simpsons generated from random vectors using the DCGAN methodology
  - “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”, https://arxiv.org/pdf/1511.06434.pdf

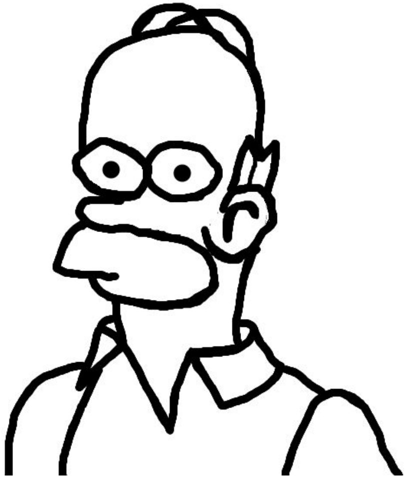
GAN Examples

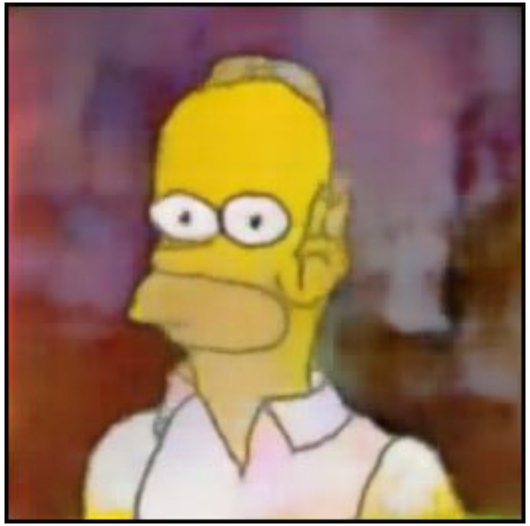


"Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks", https://arxiv.org/pdf/1511.06434.pdf

GAN Examples

- Simpsons generated from my poorly drawn sketch using the ‘Pix2Pix’ methodology
    - “Image-to-Image Translation with Conditional Adversarial Networks”, https://arxiv.org/pdf/1611.07004.pdf





GAN Examples

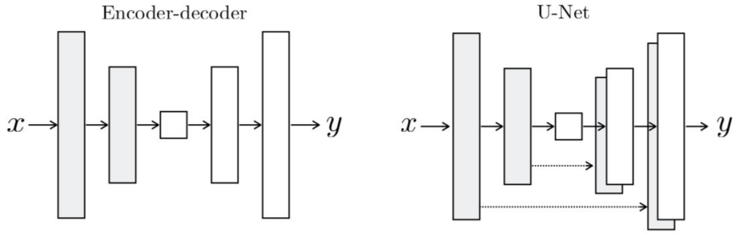
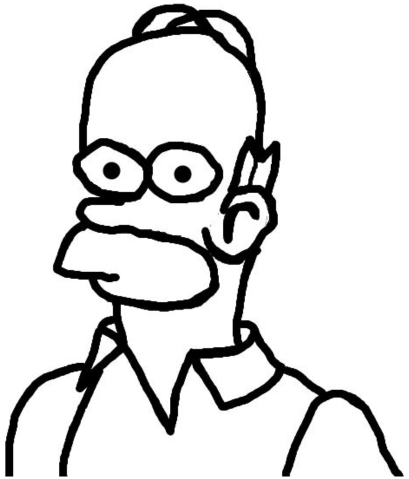
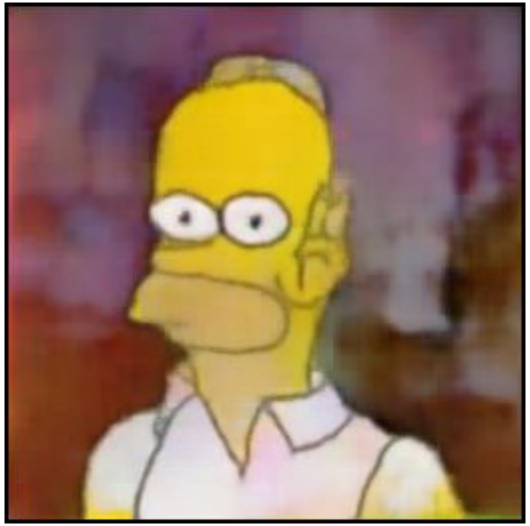


Figure 3: Two choices for the architecture of the generator. The "U-Net" [49] is an encoder-decoder with skip connections between mirrored layers in the encoder and decoder stacks.





"Image-to-Image Translation with Conditional Adversarial Networks", https://arxiv.org/pdf/1611.07004.pdf

TF Notebook - Working with Inference Graphs

Project

Project

- Use a frozen inference graph that we created during one of the previous exercises to create an application that uses a deep learning model to process an image or text.
    - Handwritten digit recognition
    - Fashion classification
    - English to pig-latin translation