Intro:

* (**Slide 1**) Generate new, synthetic data.
* (**Slide 4**) Unsupervised learning. Learn the structure of the data.
* (**Slide 5**) A model that learns to represent the underlying distribution of the data.
* (**Slide 6**) Why: Debiasing through diversity, (**Slide 7**) Outlier detection
* (**Slide 8**) Two main types: Autoencoders and GANs
  + Latent variable models. Break down into core parts and bring them to the surface.

Autoencoders:

* (**Slide 12**) Learns a lower dimensional latent space in raw data
  + Basically, the first part of a CNN. The encoder.
* (**Slide 13**) Use a decoder to bring these latent variables to the surface via reconstruction
* (**Slide 14**) The loss is the squared difference between the original and the new image
* (**Slide 16**) The lower the dimensionality, the less quality in reconstruction. Reconstructing from less information.

Variational Autoencoders:

* (**Slide 19**) Regular autoencoders are deterministic. If the weights are the same, if you feed in an input, you will get the same output.
* (**Slide 20**) VAEs use a stochastic process. The goal is to generate smoother representations. Produce reconstructions that are similar to the input, but not exact copies.
* (**Slide 21**) Use a stochastic sampling operation. Learns a mean and variance for each latent variable.
  + These parameterize a probability distribution for the latent variable.
* (**Slide 21**) Gone from a latent variable vector to a vector of means and a vector of variances. Define probability distributions for each of the latent variables.
  + Generate new samples by sampling these distributions.
* (**Slide 22**) Encoder learns a probability distribution of the latent space. Decoder computes a new distribution given the latent space.
  + Different sets of weights.
* (**Slide 23**) Loss is a function of data and sets of weights. Two terms.
  + Reconstruction loss captures difference between input and output.
  + Regularization term
* (**Slide 25**) Similar reconstruction loss
* (**Slide 26**) Regularization term:
  + Enforces a prior on the latent space during training. Enforce the latent space to follow the prior.
  + Considers the divergence between the latent space and the prior.
* (**Slide 27**) Common choice for prior is standard normal (Gaussian)
* (**Slide 29**) Yields continuity and completeness in regularization
  + Without regularization, points close in latent space may not be treated similarly and some may be meaningless.
  + (**Slide 30**) All latent variables are going to try and have a centered mean and regularized variances. Ensures smoothness and overlap in latent space.
* (**Slide 31**) Enforces the information gradient in the latent space
  + Tradeoff: More regularization may lead to less quality in the output.
* (**Slide 33**) Cannot backpropagate through stochastic sampling layer.
  + Backpropagation requires deterministic layers.
* (**Slide 34**) Solution: reparametrize the sampling layer.
  + Fixed mean vector and a fixed variance vector, which is scaled by random constant drawn from the prior distribution.
* (**Slide 36**) The latent variable is no longer a stochastic layer, but it takes input from one.
* (**Slide 37**) Now we can edit a single latent variable to see what it does
* (**Slide 38**) To have uncorrelated latent variable, we introduce disentanglement
  + Learn the best latent representation possible.
* (**Slide 39**) Introduce a beta term, which controls the strength of regularization.
  + Places constraints on the latent encoding to encourage disentanglement.
  + In the left, changing one latent variable changes others. On the right, changing one latent variable only.
* (**Slides 41-45**) Summary

GANs:

* (**Slide 46**) Goal is to generate samples that are as close to the original data as possible
* (**Slide 47**) Don’t want to model the density behind the data. Just to create new data.
  + Start from noise and learn a transformation to the data distribution.
* (**Slide 48**) GANs (insert joke), utilize two networks. A discriminator and a generator.
  + These two networks compete. They are adversaries.
  + The generate starts with noise and tries to create an image as close to the real data as possible.
  + The discriminator looks at the real images and the generated images to determine which is fake.
  + During training, this forces the generator to improve itself and create better synthetic data. Moves its synthetic data closer to the distribution of the real data.
  + The generator starts out poor and gets better. The discriminator starts great and gets poor.
* (**Slide 67**) Loss: we have to adversarial objectives for the discriminator and the generator.
  + The end result is the generator producing the best synthetic data it can, close to the true data distribution.
* (**Slide 68**) Discriminator:
  + Maximize the probability that the fake data is identified as fake.
  + G(z) is the generator’s output.
  + D(G(z)) is the probability that a fake instance is fake.
  + D(x) is the probability that a real input is fake.
  + 1-D(x) is the probability that a real input is real.
  + Maximize the probability that fake is fake, real is real.
* (**Slide 69**) Generator:
  + Has the adversarial objective to the discriminator.
  + Minimize the probability that the discriminator knows what is real and what is fake.
  + Minimize the probability that fake is fake, real is real.
* (**Slide 70**) The loss function:
  + Min max objective function.
* (**Slide 71**) After training we use the generator to generate new data.
* (**Slide 72**) The transformation from noise to a target distribution is what is learned during training
  + One point in the noise results in a particular output.
  + (**Slide 73**) A different point, another
* (**Slide 74**) We can traverse through the noise to see the traverse in the data distribution location.

Advanced GANs:

* (**Slide 76**) Progressive GANs
  + Build more detail in output.
  + Progressively adding layers of increasing spatial resolution.
* (**Slide 78**) StyleGAN
  + Combines progressive growth with style transfer
    - Compose an image in the style of another.
  + Source A mimics the style of source B.
* (**Slide 81**) Conditional GANs
  + Condition on a label using conditioning factor.
  + (**Slide 82**) Allows for paired translation
    - We now have pairs of inputs.
* (**Slide 86**) CycleGAN
  + Domain transformation.
  + Learns a mapping from one domain to apply to another. Transfer the style and distribution.
  + Go back and forth between a domain X and domain Y.
  + There are two generators and two discriminators.
  + (**Slide 87**) Instead of moving from noise to data, we are moving from data to data.
* (**Slide 90**) Summary: