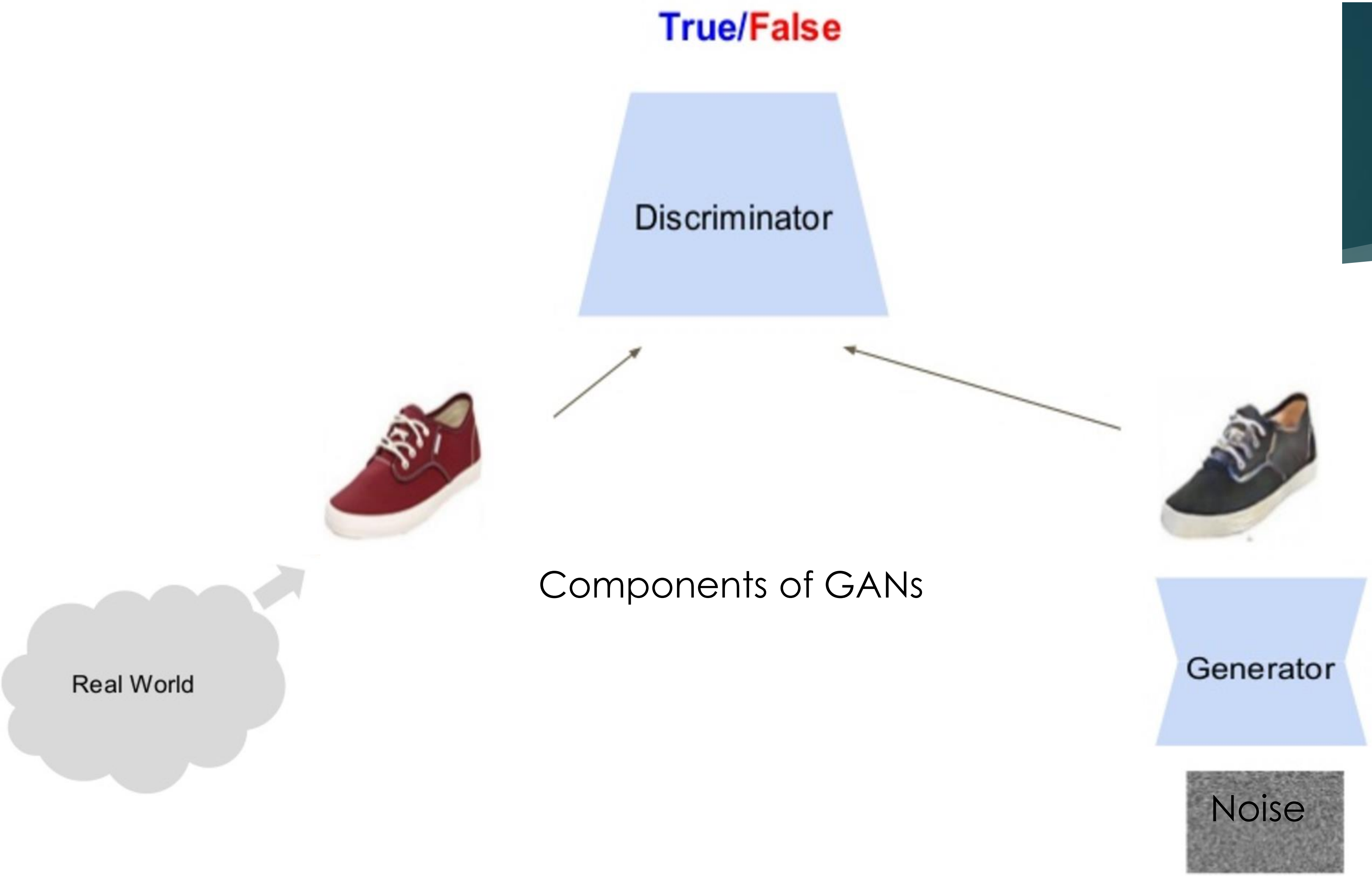
Generative Adversarial Networks

AKRIT MOHAPATRA

ECE DEPARTMENT, VIRGINIA TECH

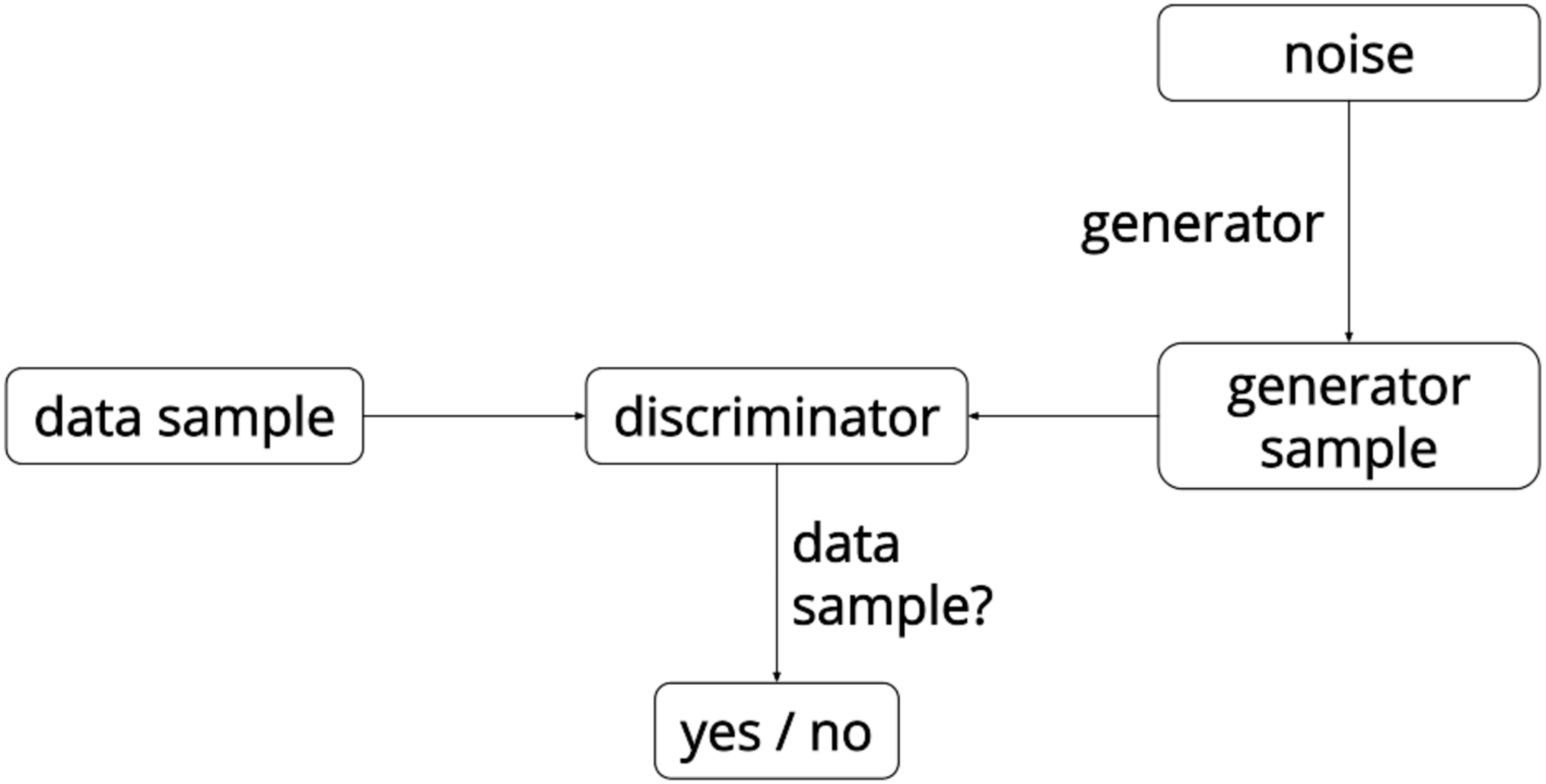
What are GANs?

- System of two neural networks competing against each other in a zero-sum game framework.
- They were first introduced by Ian Goodfellow *et al.* in 2014.
- Can learn to draw samples from a model that is similar to data that we give them.



Components of GANs

Slide credit – Victor Garcia



Overview of GANs
Source: https://ishmaelbelghazi.github.io/ALI

Discriminative Models

- A **discriminative** model learns a function that maps the input data (x) to some desired output class label (*y*).
- In probabilistic terms, they directly learn the conditional distribution $P(y|x)$.

Generative Models

- A **generative** model tries to learn the joint probability of the input data and labels simultaneously i.e. $P(x,y)$.

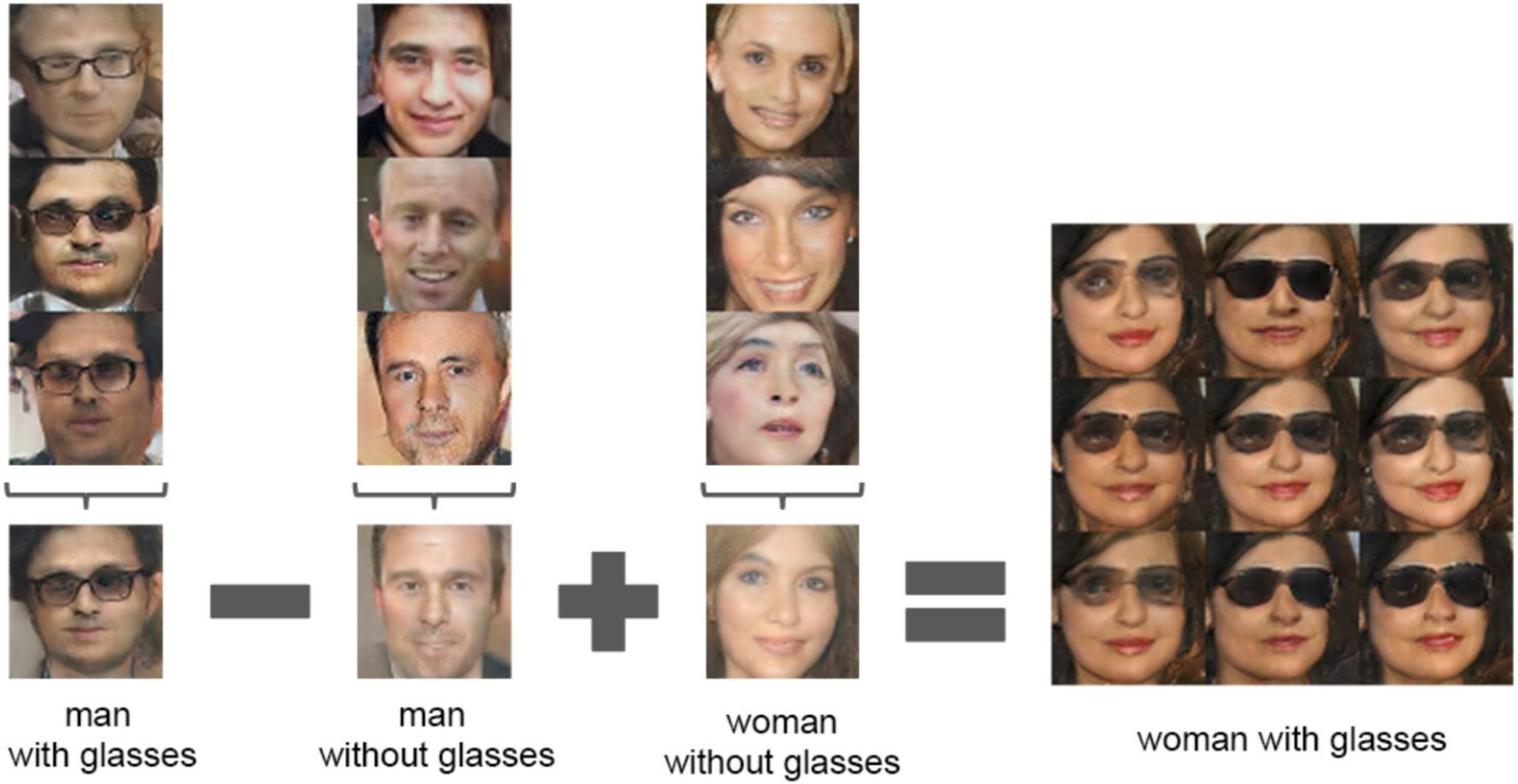
- Potential to understand and explain the underlying structure of the input data even when there are no labels.

How GANs are being used?

- Applied for modelling natural images.
- Performance is fairly good in comparison to other generative models.
- Useful for unsupervised learning tasks.

Why GANs?

- Use a latent code.
- Asymptotically consistent (unlike variational methods) .
- No Markov chains needed.
- Often regarded as producing the best samples.


man
with glasses
man
without glasses
woman
without glasses
woman with glasses

How to train GANs?

- Objective of generative network - increase the error rate of the discriminative network.
- Objective of discriminative network – decrease binary classification loss.
- Discriminator training - backprop from a binary classification loss.
- Generator training - backprop the negation of the binary classification loss of the discriminator.

Loss Functions

$$\mathcal{L}(\hat{x}) = \min_{x \in data} (x - \hat{x})^2$$

Generator

$$D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}$$

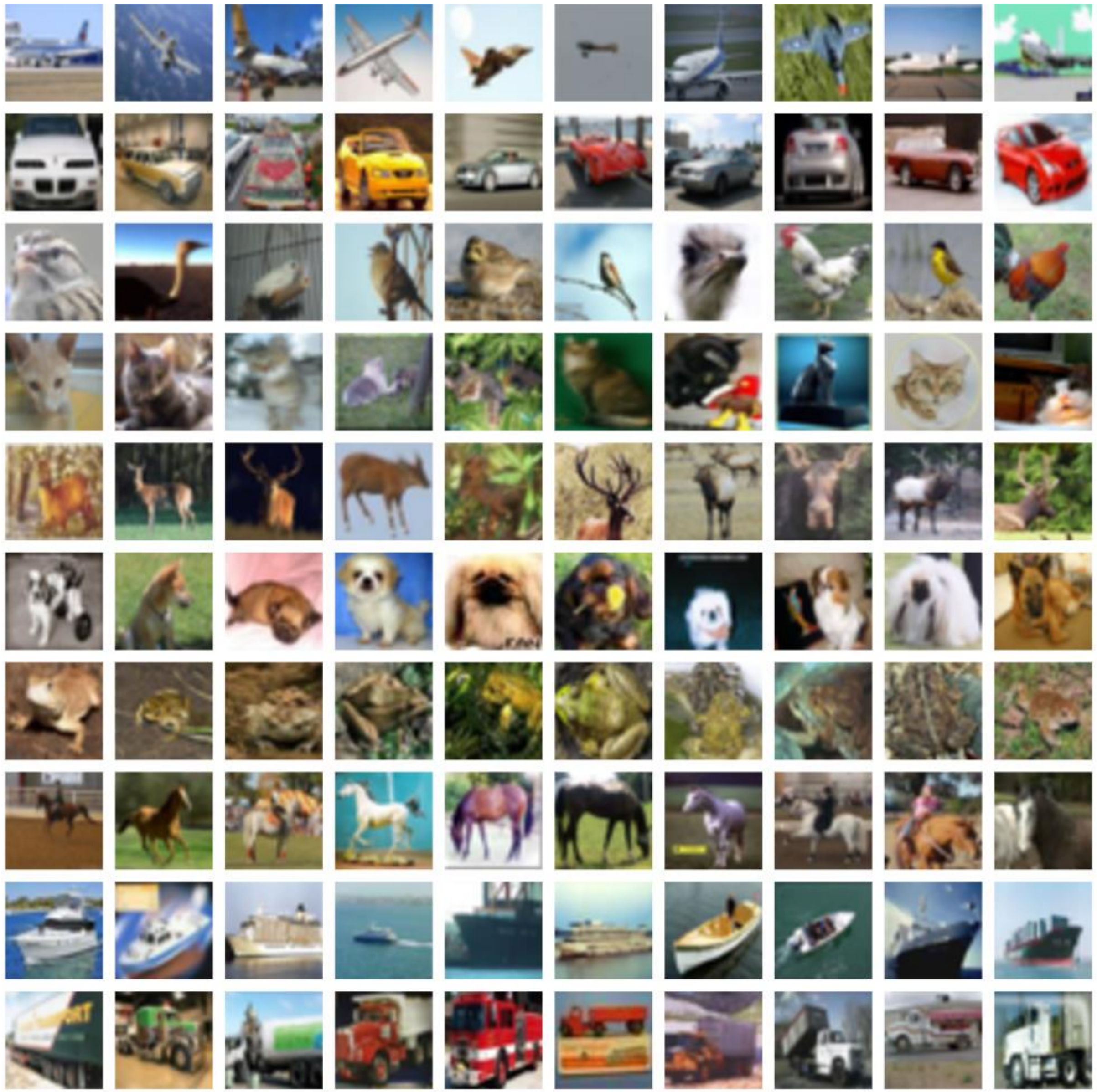
Discriminator

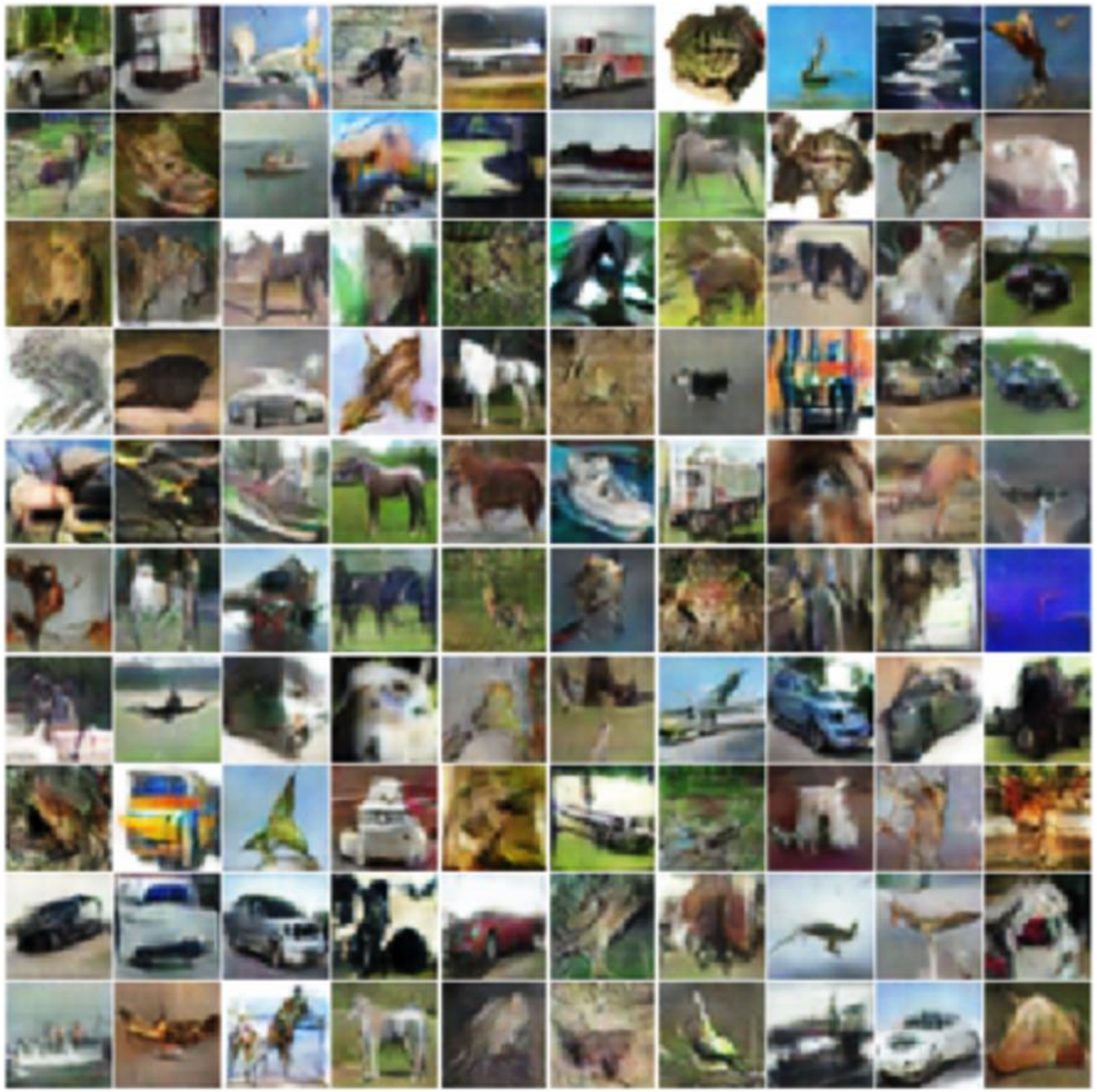
Generated bedrooms. Source: "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks" https://arxiv.org/abs/1511.06434v2

"Improved Techniques for Training GANs" by Salimans et. al

- One-sided Label smoothing - replaces the 0 and 1 targets for a classifier with smoothed values, like .9 or .1 to reduce the vulnerability of neural networks to adversarial examples.

- Virtual batch Normalization - each example x is normalized based on the statistics collected on a reference batch of examples that are chosen once and fixed at the start of training, and on x itself.





Original CIFAR-10 vs. Generated CIFAR-10 samples

Source: "Improved Techniques for Training GANs" https://arxiv.org/abs/1606.03498

Variations to GANs

- Several new concepts built on top of GANs have been introduced –
  - InfoGAN – Approximate the data distribution and learn interpretable, useful vector representations of data.
  - Conditional GANs - Able to generate samples taking into account external information (class label, text, another image). Force G to generate a particular type of output.

Major Difficulties

- Networks are difficult to converge.
- Ideal goal – Generator and discriminator to reach some desired equilibrium but this is rare.
- GANs are yet to converge on large problems (E.g. Imagenet).

Common Failure Cases

- The discriminator becomes too strong too quickly and the generator ends up not learning anything.
- The generator only learns very specific weaknesses of the discriminator.
- The generator learns only a very small subset of the true data distribution.

So what can we do?

- Normalize the inputs
- A modified loss function
- Use a spherical Z
- BatchNorm
- Avoid Sparse Gradients: ReLU, MaxPool
- Use Soft and Noisy Labels
- DCGAN / Hybrid Models
- Track failures early (D loss goes to 0: failure mode)
- If you have labels, use them
- Add noise to inputs, decay over time

Conclusions

- Train GAN – Use discriminator as base model for transfer learning and the fine-tuning of a production model.

- A well-trained generator has learned the true data distribution well - Use generator as a source of data that is used to train a production model.

Dive Deeper?

Generative Adversarial Networks (GANs)

Ian Goodfellow, OpenAI Research Scientist
NIPS 2016 tutorial
Barcelona, 2016-12-4

OpenAI

Ian Goodfellow's NIPS 2016 Tutorial

Available online.

References

- https://tryolabs.com/blog/2016/12/06/major-advancements-deep-learning-2016/
- https://blog.waya.ai/introduction-to-gans-a-boxing-match-b-w-neural-nets-b4e5319cc935#.6l7zh8u50
- https://en.wikipedia.org/wiki/Generative_adversarial_networks
- http://blog.aylien.com/introduction-generative-adversarial-networks-code-tensorflow/
- https://github.com/soumith/ganhacks