A brief Overview On Generative Adversarial Networks (GANs)

Supervised by:
Prof.Abdul Qayyum

Contributors:
Adem Sağlam, Thien Bao Bui & Syed Muhammad Hashaam Saeed

Date: 17/12/2020

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.

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.

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 $P(y|x)$.

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.

How to train GANs?

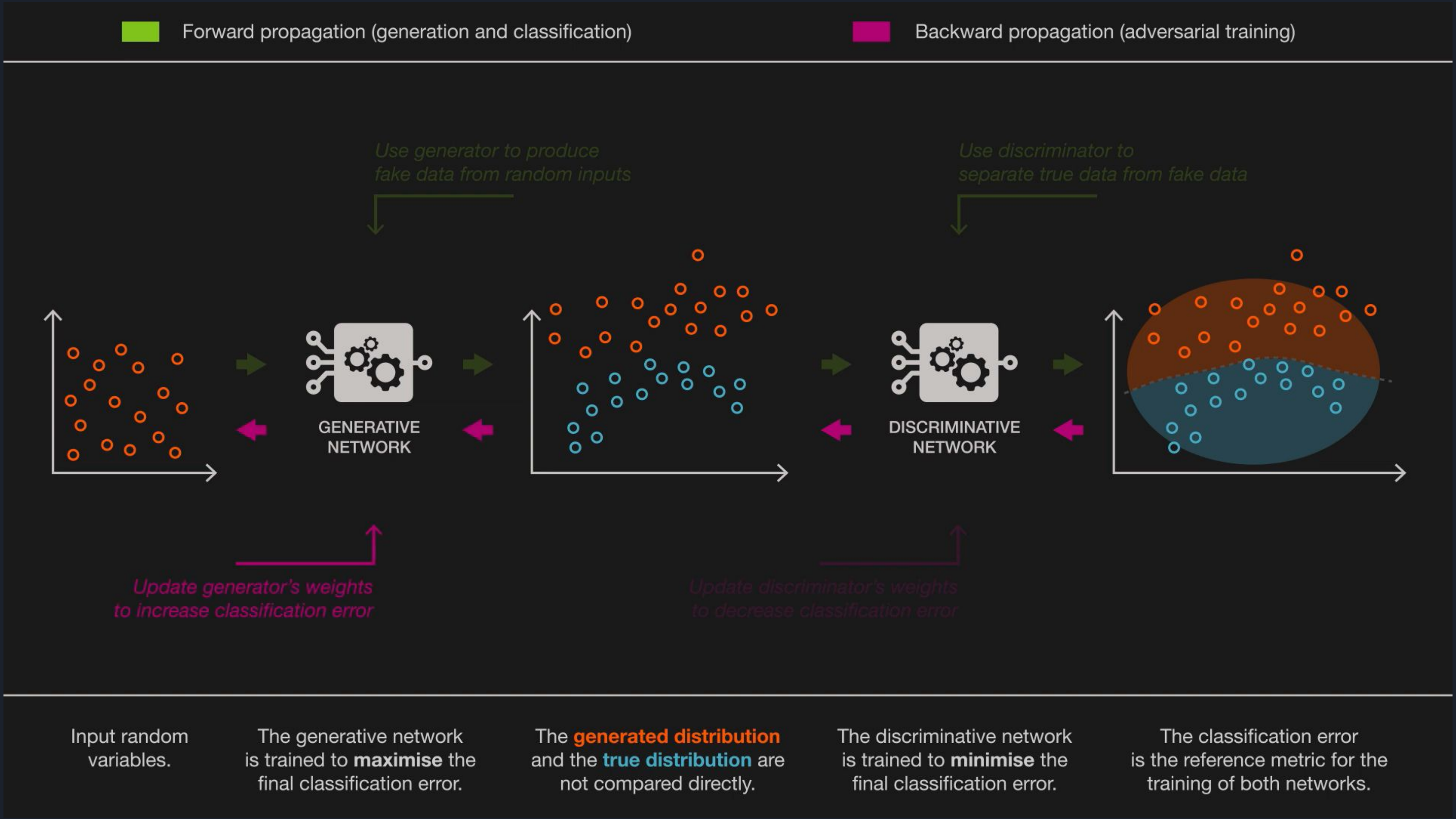
- Objective of generative network - increase the error rate of the discriminative network.
- Objective of discriminative network – decrease binary classification loss.
- Discriminator training - backprop from a binary classification loss.
- Generator training - backprop the negation of the binary classification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

GAN Loss

$$V(D,G) = E_x[log(D(x))] + E_z[log(1 - D(G(z)))]$$

- In the paper that introduced GANs, the generator tries to minimize the function above while the discriminator tries to maximize it , it's called the min-max loss

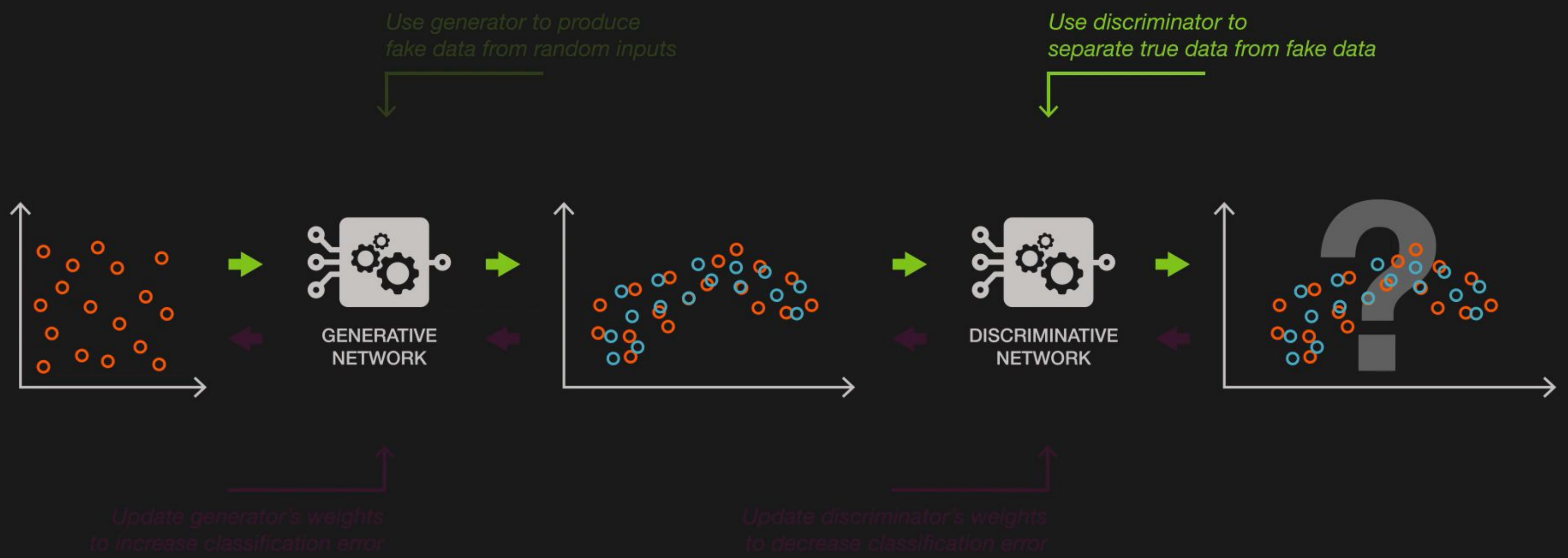
- The generator can't directly affect the log(D(x)) term in the function, so, for the generator, minimizing the loss is equivalent to minimizing log(1 - D(G(z))).



Forward propagation (generation and classification)
Backward propagation (adversarial training)
Use generator to produce fake data from random inputs
Use discriminator to separate true data from fake data
GENERATIVE NETWORK
DISCRIMINATIVE NETWORK
Update generator's weights to increase classification error
Update discriminator's weights to decrease classification error
Input random variables.
The generative network is trained to maximise the final classification error.
The generated distribution and the true distribution are not compared directly.
The discriminative network is trained to minimise the final classification error.
The classification error is the reference metric for the training of both networks.

Forward propagation (generation and classification)

Backward propagation (adversarial training)



Input random variables.

The generative network is trained to **maximise** the final classification error.

The **generated distribution** and the **true distribution** are not compared directly.

The discriminative network is trained to **minimise** the final classification error.

The classification error is the reference metric for the training of both networks.

“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.

How to maximise likelihood of convergence?

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

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.

Opportunities in medical imaging

Fundamental problems in medical imaging field:

- Lack of labeled data

- Class imbalance

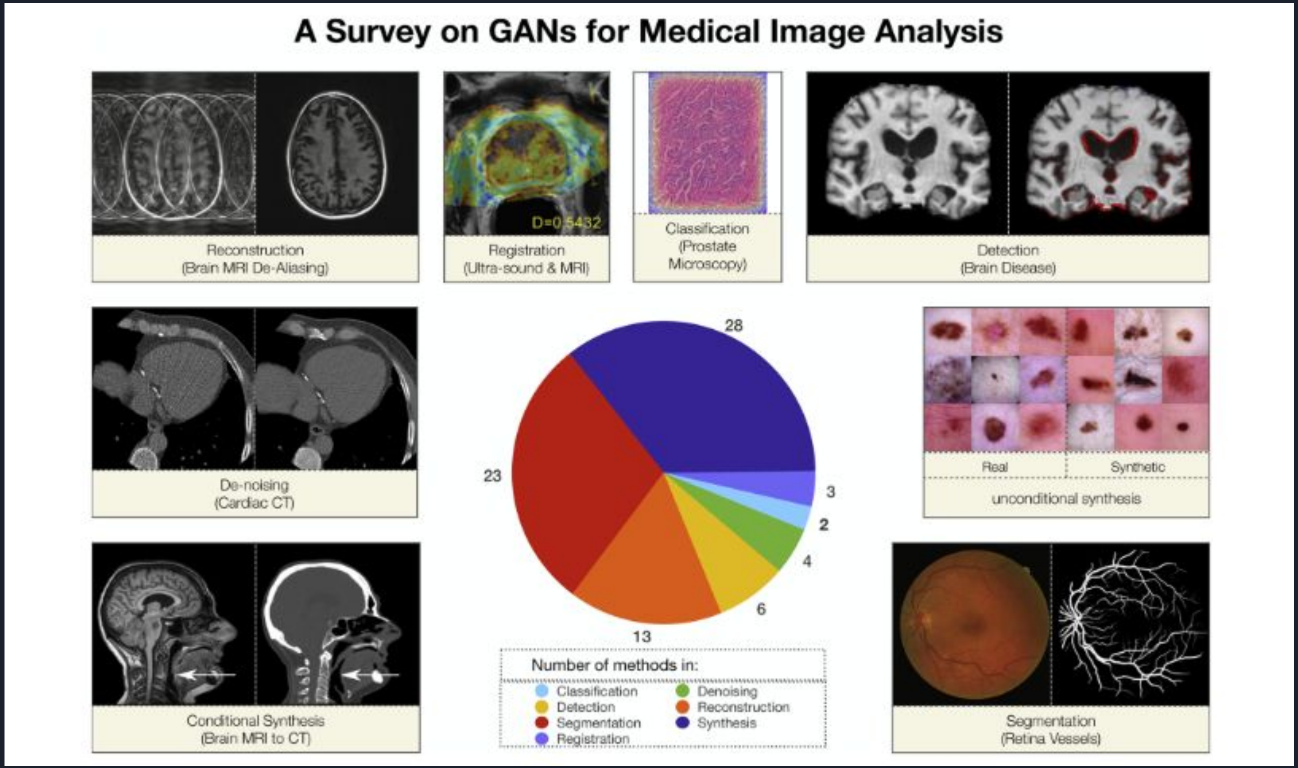
  => GANs can solve this by generating realistic looking images from an implicit distribution that follows the real data distribution

Opportunities in medical imaging

Applied research on GANs for this field can be classified in 2 types:

- Unconditional image synthesis
- Conditional image synthesis

Applications of GANs in Medical Imaging



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.

References

- GANs for medical image analysis: https://doi.org/10.1016/j.artmed.2020.101938