Generative Adversarial Nets

Short presentation of the paper by Goodfellow et al.

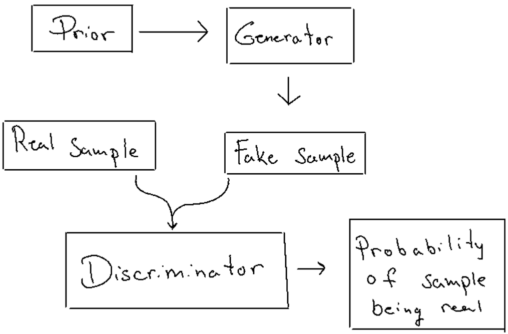
Makito Fredskild Katsume

UNIVERSITY OF COPENHAGEN

What is a generative adversarial network?

- Two models trained simultaneously, the generator and the discriminator.
- A minmax game played between the generator and the discriminator.

$$min_G max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{data}(\boldsymbol{x})}[log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_z(\boldsymbol{z})}[log(1 - D(G(\boldsymbol{z})))]$$



A graphical view

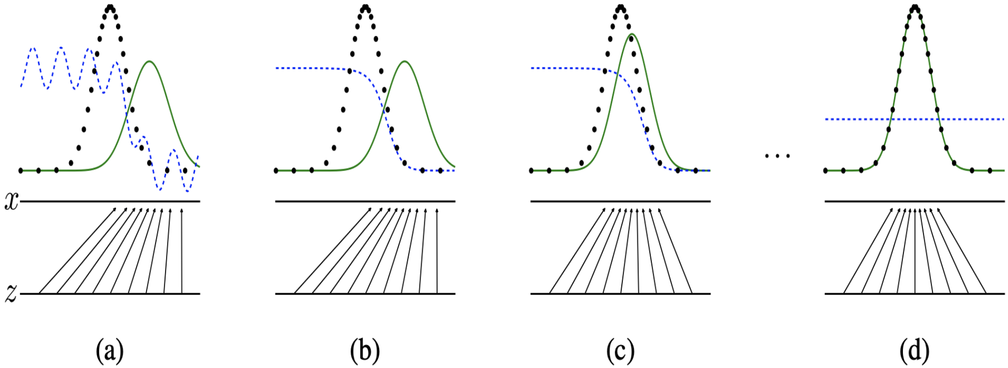


Figure: From "Generative adversarial nets" by Goodfellow et al.[1]

Algorithm

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**
  for k steps **do**
    • Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
    • Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
    • Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)\right].$$

  end for
  • Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  • Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for
The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

Figure: From "Generative adversarial nets" by Goodfellow et al.[1]

Theoretical result, the short version

Given:

- Models have infinite capacity and training time
- During each step in the training the discriminator finds the optimum for a given generator

Then:
The distribution of the generator converges to the distribution of the training data[1].

My application

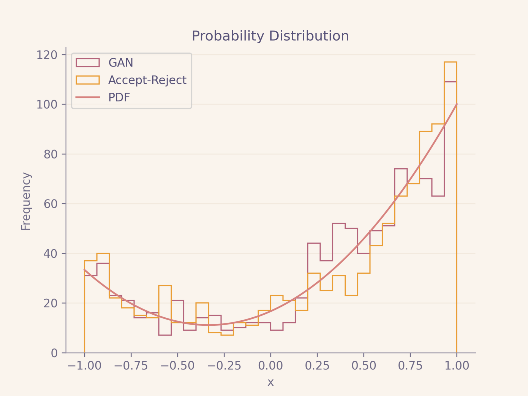
The distribution to be approximated is

$$pdf(x) = C(1 + \alpha x + \beta x^2) \text{ for } x \in [-1, 1]$$

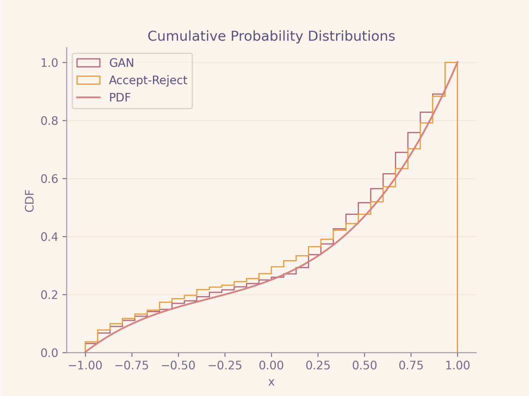
where C is the normalisation constant, $\alpha = 2$, and $\beta = 3$.

- Prior space uniformly distributed on $\boldsymbol{z} \in [0, 1]$
- Real data is sampled by Accept-Reject method

Result

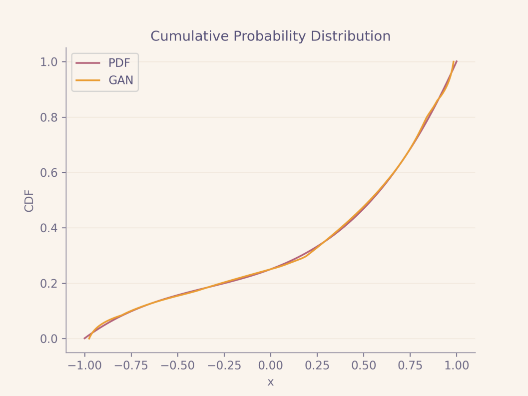


(a) A comparison of samples generated with Accept-Reject method and the GAN model.

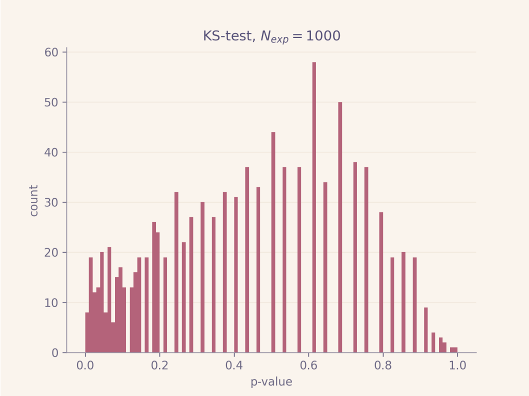


(b) Cumulative distribution of the sample from the left plot.

Result



(a) Cumulative distribution of the resulting GAN and the training PDF.



(b) KS-Test on 1000 iteration.

Conclusion

- It kind of works.
- A lot of meta parameters to choose and adjust.
- Needs a lot of labelled data.
- Could be viable in the right use case.

Reference **:**

[1] Ian J. Goodfellow et al. "Generative Adversarial Nets". In: (2014). DOI: https://arxiv.org/abs/1406.2661.