Introduction

While increased success in methods in deep learning has given rise to very robust discriminative models, their success with deep generative models was very limited owing to the "approximation of many intractable probabilistic computations that arise in maximum likelihood estimation and related strategies", as is described in the paper 'Generative Adversarial Nets' by I. J. Goodfellow et. al[1]. This paper introduces a novel strategy which revolutionized the field of generative modeling when it was published.

The main idea is that a generative model battles out an adversary, which in this case is a discriminative model. The generative model attempts to churn out new data samples through a prior noise distribution. This data is fed into the discriminative model. The discriminative model attempts to classify if this data originated from model distribution, or the data distribution. Both of these models take feedback from each other.

Design of the model

The generative model passeses random noise through its neural network to produce new samples. The discriminative model, also a multi layer perceptron, attempts a traditional classification task. In this case, there are only 2 possible labels- sample generated from the generative model or sample sampled from the data distribution.

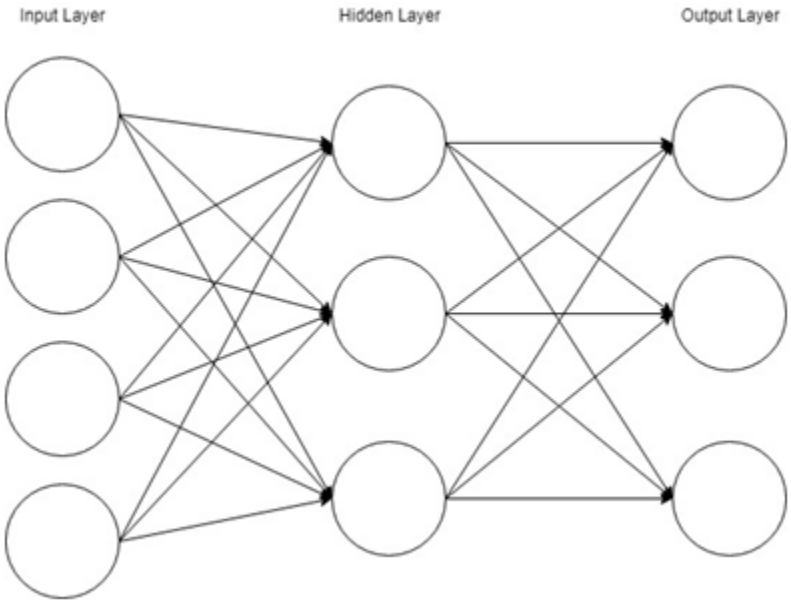

Fig. 1 Diagramatic representation of an MLP
Image credits: https://deepai.org/machine-learning-glossary-and-terms/multilayer-perceptron

The generative model inputs random noise through the input layer, and generates new samples. If the generative model is being trained to produce pictures, each perceptron in the output layer could correspond to a pixel, encoding its color data.

The Minimax Game

Let us define the following notation:

p_g - Generator's distribution

$p_z(z)$ - The prior on input noise variables

$G(z;\theta_g)$ - The generative model which takes in the noise and maps it to the data space,comprising of the parameters θ_g .

$D(x;\theta_d)$ - The discriminative model which outputs a single scalar, representing the probability that the **x** came from the data rather than p_g .

The adversaries(generative and discriminative model) now participate in the following minimax game.

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

This forms the main crux of the paper. Do note that this equation is immediately very intuitive. For D to maximize this equation, it would need to label all of the data correctly. That means D(x) will evaluate to 1, and D(G(z)) will evaluate to 0, since this sample was generated from the generative model.

Similarly, when the equation is minimized, the second term is minimized, and that would mean that $1 - D(G(z))$ tended to zero. That means at this stage the discriminative model was incorrectly classifying the data generated from the generative model as 'real' data- exactly what the generative model wanted from the start.

While this equation is very intuitive, it needs to be tweaked to work in practice. More specifically, $log(1 - D(G(z))$ does not provide sufficient gradient in the beginning to learn well, since the samples from the generative models would clearly differ from the training data. Training G to maximize $D(G(z))$ proves to be much better in practice.

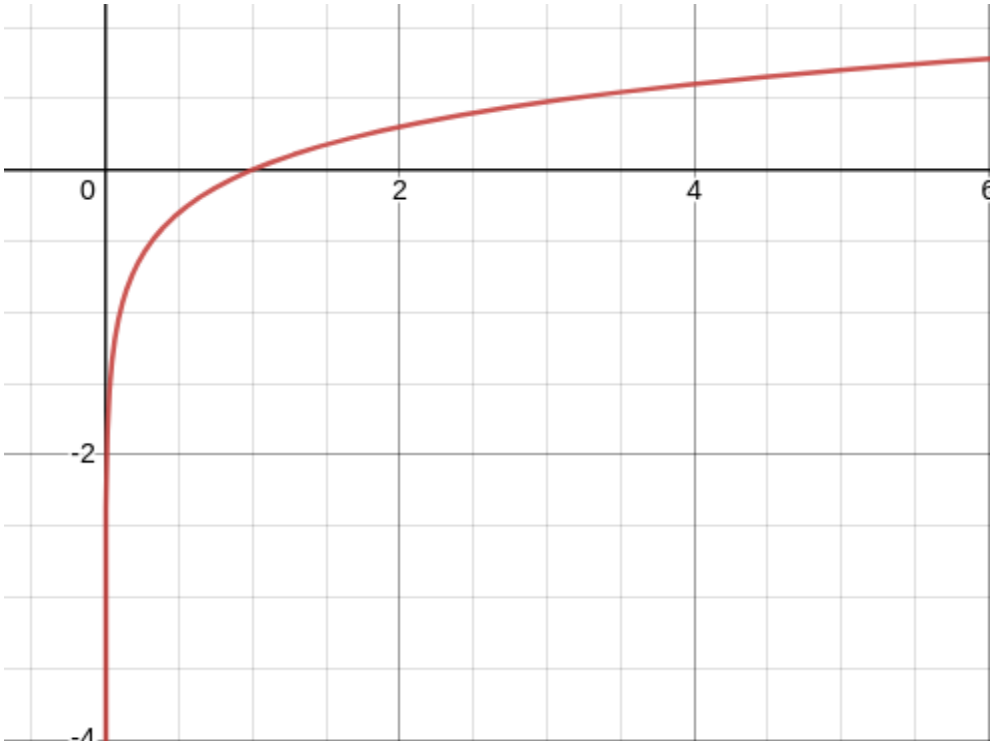

Fig. 2 Plot of log(x) vs x

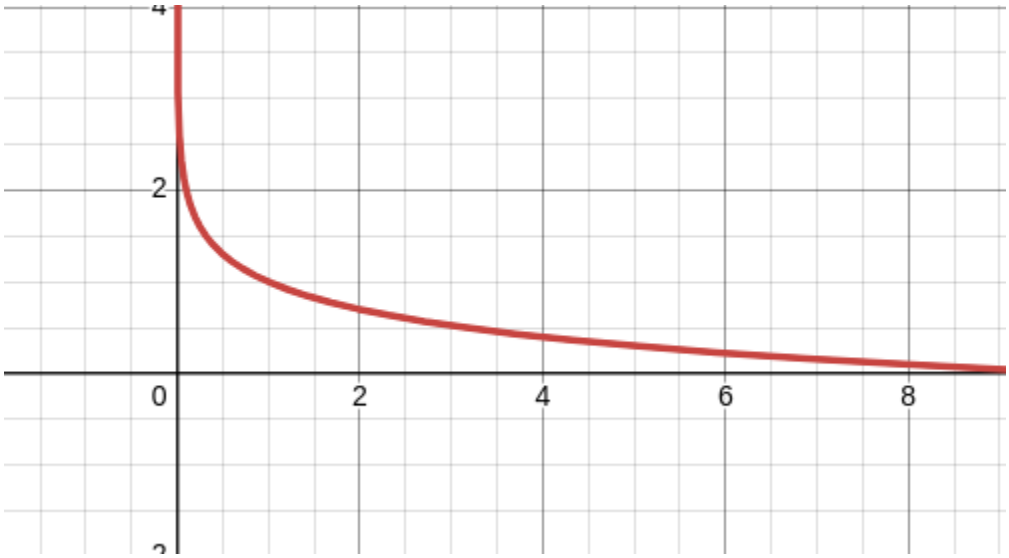

Fig. 3 Plot 1-log(x) vs x

As is visible from Fig. 2 and Fig. 3, the slope of the graph at x = 0 in Fig. 1 is much more steep than the slope at x = 1 in Fig. 2.

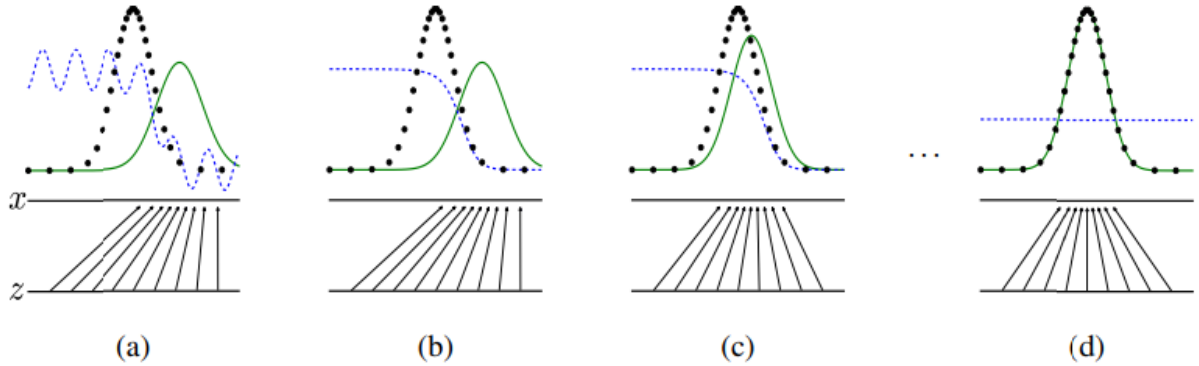

Fig. 4 Evolution of D and G with training. The blue dotted lines represent the output from D, the solid green line represents the data distribution and the dotted black line represents the model distribution.
Image credits: https://arxiv.org/pdf/1406.2661.pdf

Fig. 4 is an excellent graph from the paper depicting the evolution of the adversaries. At the start, D does not classify well and the model distribution differs a lot from the data distribution. As D is trained, it starts to correctly classify samples from the data and samples from G. As training is progressed further, samples from G become indistinguishable from the data samples.

Theoretical Discussion

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 $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
    • Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ 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 $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ 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.

Fig. 5 Pseudocode of the algorithm
Image credits : https://arxiv.org/pdf/1406.2661.pdf

The first proposition describes the optimal discriminator D for a fixed G.

Proposition 1. *For G fixed, the optimal discriminator D is*

$$D_G^*(\boldsymbol{x}) = \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_g(\boldsymbol{x})} \tag{2}$$

Proof. The training criterion for the discriminator D, given any generator G, is to maximize the quantity $V(G, D)$

$$\begin{aligned} V(G,D) &= \int_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \log(D(\boldsymbol{x}))dx + \int_z p_{\boldsymbol{z}}(\boldsymbol{z}) \log(1 - D(g(\boldsymbol{z})))dz \\ &= \int_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \log(D(\boldsymbol{x})) + p_g(\boldsymbol{x}) \log(1 - D(\boldsymbol{x}))dx \end{aligned} \tag{3}$$

For any $(a, b) \in \mathbb{R}^2 \setminus \{0, 0\}$, the function $y \to a \log(y) + b \log(1 - y)$ achieves its maximum in $[0, 1]$ at $\frac{a}{a+b}$. The discriminator does not need to be defined outside of $Supp(p_{\text{data}}) \cup Supp(p_g)$, concluding the proof. □

As a result, the minimax game now becomes:

$$\begin{aligned} C(G) &= \max_D V(G,D) \\ &= \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}}[\log D_G^*(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}}[\log(1 - D_G^*(G(\boldsymbol{z})))] \\ &= \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}}[\log D_G^*(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x} \sim p_g}[\log(1 - D_G^*(\boldsymbol{x}))] \\ &= \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}}\left[\log \frac{p_{\text{data}}(\boldsymbol{x})}{P_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right] + \mathbb{E}_{\boldsymbol{x} \sim p_g}\left[\log \frac{p_g(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right] \end{aligned}$$

The first theorem states that:
The global minimum of the virtual training criterion C(G) is achieved if and only if $p_g = p_{data}$.
At that point, C(G) achieves the value − log 4.

To prove the following theorem, add and subtract log 4 from the above expression. After some refactoring, we get

$$C(G) = -\log(4) + KL\left(p_{\text{data}} \middle\| \frac{p_{\text{data}} + p_g}{2}\right) + KL\left(p_g \middle\| \frac{p_{\text{data}} + p_g}{2}\right)$$

Where KL is the Kullback-Leibler divergence.

This evaluates to

$C(G) = -\log(4) + 2 \cdot JSD\left(p_{\text{data}} \| p_g\right)$, where JSD is the Jensen-Shannon divergence. Since the minimum value it can take is 0, the global optimum for C(G) is indeed -log 4.

The paper also presents a proof for the fact that if G and D have enough capacity, and the discriminator D takes its optimum for every G, then p_g converges to p_{data}.

Conclusion

This paper proved to be a seminal work in the field of generative modeling. For many years after it was published, it set the direction for future research in the field. The main thing to takeaway is the implementation of two MLPs, each trying to achieve their own optimum. In this process, the generative model grows to generate very realistic samples.

Suggestions

Reproduction of the results in the paper could be a difficult task, since there is very little discussion on the implementation of the neural network like their architecture and the hyperparameters used.

Moreover, although some disadvantages are stated briefly(like mode collapse), they need to be discussed in further detail and possibly provide some direction to solving these issues.

These are very minute problems in an otherwise brilliant paper, and do not really take away from the experience of reading and understanding the paper.

References

[1] https://arxiv.org/pdf/1406.2661.pdf