Notes on Generative Adversarial Nets

January 26, 2022

1 Related Work

- Undirected graphical models with latent variables
  - Restricted Boltzmann Machines
  - Deep Boltzmann Machines
- Deep Belief Networks
- Noise-Contrastive Estimation
- Generative Stochastic Network
- Variational Auto Encoders
- Stochastic Backpropagation

2 Adversarial Nets

- Goal: Learn distribution p_g over data x.
- Method:
  - Prior on input variables $p_z(z)$
  - Map $z \to G(z; \theta_g)$. G is differentiable.
  - Map $x \to D(x; \theta_d)$
  - Train:
    * D to mazimize the probability of classifying correctly its inputs (real or fake).
    * G to minimize $\log(1 - D(G(z)))$
    * In short: $\min_G \max_D V(G, D) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]$
    * It is preferable to rain G by maximizing $\log(D(G(z)))$ to obtain stronger gradients at the beginning (by avoiding saturation).
  - After several steps, the equilibrium (point where neither can make improvements) will be reached and $p_g = p_{data}$ (given that both networks have enough capacity).

3 Theoretical Results

This assumes that models have infinite capacity in order to study convergen in the space of probability density functions.

For G fixed, the optimal discriminator D is
$D^*_G(x) = \frac{p_{data}(x)}{p_{data}(x)+p_g(x)}$

$$
\begin{aligned}
V(G, D) &= \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \\
&= \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{x\sim p_g(x)}[\log(1 - D(x))] \\
&= \int_x p_{data}(x) \log(D(x)) + p_g(x) \log(1 - D(x)) dx
\end{aligned}
$$

$(a, b) \in \mathbb{R}^2 \setminus \{0, 0\},\ y \rightarrow a\log(y) + b\log(1-y)$ achieves its maximum in $[0, 1]$ at $\frac{a}{a+b}$ (derivative w.r.t. y and make it $= 0$). The discriminator does not need to be defined outside of $Supp(p_{data}) \cup Supp(p_g)$ i.e. when $(a, b) = (0, 0)$

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

According to this last bit, D's training objective can be interpreted as maximizing the log-likelihood for estimating the conditional probability $P(Y = y|x)$ where Y is the random variable that indicates whether x belongs to $p_{data}(y = 1)$ or to $p_g(y = 0)$.

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 $-\log(4)$.

$$
\begin{aligned}
C(G) &= \mathbb{E}_{x\sim p_{data}}\left[\log p_{data}(x) - \log\frac{p_{data}(x) + p_g(x)}{2} - \log 2\right] + \mathbb{E}_{x\sim p_g}\left[\log p_g(x) - \log\frac{p_{data}(x) + p_g(x)}{2} - \log 2\right] \\
&= -\log(4) + \mathbb{E}_{x\sim p_{data}}\left[\log p_{data}(x) - \log\frac{p_{data}(x) + p_g(x)}{2}\right] + \mathbb{E}_{x\sim p_g}\left[\log p_g(x) - \log\frac{p_{data}(x) + p_g(x)}{2}\right] \\
&= -\log(4) + KL\left(p_{data}\middle|\middle|\frac{p_{data} + p_g}{2}\right) + KL\left(p_g\middle|\middle|\frac{p_{data} + p_g}{2}\right) \\
&= -\log(4) + 2JSD(p_{data}||p_g)
\end{aligned}
$$

If G and D have enough capacity, and at each step of the Algorithm 1, the discriminator is allowed to reach its optimum given G, and p_g is updated so as to improve the criterion

$$
\mathbb{E}_{x\sim p_{data}}[\log D^*_G(x)] + \mathbb{E}_{x\sim p_g}\left[\log(1 - D^*_G(x))\right]
$$

then p_g converges to p_{data}.

Let $V(G, D) = U(p_g, D)$.

- Note that $U(p_g, D)$ is convex in p_g. Why?
  - The subderivatives of a supremum of convex functions include the derivative of the function at the point where the maximum is attained.
    * In other words: If $f(x) = \sup_{\alpha\in A} f_\alpha(x)$ and $f_\alpha(x)$ is convex in x for every α, then $\partial f_\beta(x) \in \partial f$ if $\beta = \arg\sup_{\alpha\in A} f_\alpha(x)$
  - Since $\sup_D U(p_g, D)$ is convex in p_g with a unique global optima, it converges to p_x with small enough updates.

References

[1] Ian J. Goodfellow et al. *Generative Adversarial Networks*. 2014. arXiv: `1406.2661 [stat.ML]`.