CSE6243: Advanced Machine Learning **Fall 2024**

Lecture 15: Generative Adversarial Networks

Lecturer: Bo Dai *Scribes: Adithya Vasudev, Abdulaziz Memesh*

Note: *LaTeX template courtesy of UC Berkeley EECS Department.*

15.1 Recap

15.1.1 Energy Based Models

Over the past few weeks, we've been discussing generative models from the lens of an energy-based model, that is, those parameterized in the form:

$$p_f(x) = \frac{\exp(f(x))}{z(f)}$$

We then discussed contrastive divergence in energy-based models as an approximation of their MLE objective, score matching, and the trade-off between sampling and generation. We also dove into Langevin Dynamics, and their connection to various generative models, such as Diffusion, Latent Space Models, and Autoregressive Models.

15.1.2 Score Matching Review

As a refresher, score matching originates from the Langevin Dynamics sampler. Recall that in Langevin Dynamics:

$$x_{t+1} = \eta \nabla_{x_t} \log(p(x_t)) + \sqrt{\eta}\epsilon$$

Consider the first-order term, more specifically, let a **score function** $S_w(x)$ be defined as:

$$S_w(x) = \nabla_x f(x)$$

The goal is to compute the quality of this score, which is done via the KL-divergence of this score function with respect to the first-order term. We naturally want this difference minimized This is done by evaluating the following integral:

$$\int p_\lambda(x) \left\| S_w(x) - \nabla_x \log p_\lambda(x) \right\|^2 dx$$

This integral is intractable, but if we expand the norm term out, as such:

$$= \int p_\lambda(x) S_w^\top(x) S_w(x) dx + \int p_\lambda(x) \left(\nabla_x \log p_\lambda(x)\right)^2 dx - 2 \int p_\lambda(x) S_w(x) \nabla_x \log p_\lambda(x) dx$$

We can see that the first term simplifies to: $\mathbb{E}_{\hat{p}_\lambda}[S^2(w)]$ and the second term has none of the variables we are trying to optimize, making them effectively a constant and can be safely ignored (constant terms reduce to 0 when we take the first derivative).

Thus, the only problematic term is that third and final term. We can do some algebraic manipulation on this term, as such, to see of it reduces to or is bounded by a simpler term we can integrate:

$$= -2 \int p_\lambda(x) S_w(x) \frac{1}{p_\lambda(x)} \nabla p_\lambda(x) dx$$

$$= -2 \int S_w(x) \nabla p_\lambda(x) dx$$

Apply integration by parts:

$$= -2[S_w(x) p_\lambda(x)] - 2 \int p_\lambda(x) \nabla S_w(x) dx$$

That second term in this expression is the expectation of $\nabla S_w(x)$, which is our Langevin Term!

15.2 New Content

15.2.1 Generative Adversarial Networks

By their motivation, the **Generative Adversarial Network**, or GAN, is not inherently related to energy-based models, at least, not at their inception. Their goal was to solve the following minimax problem.

Define a distribution p that we are trying to learn, and let this learned distribution be q. Thus, if $x \sim p$, then let the target $y = 1$, and if $x \sim q$, then $y = -1$.

Define a function $D_\phi(x)$, called the **discriminator**, who's goal is to learn whether a value x was drawn from p or q. This is a binary classification problem, and the goal is to learn the function $D_\phi(x)$ that maximizes the score on this binary classification task.

This can be mathematically expressed as:

$$\max_{D_\phi} \mathbb{E}_{\hat{p}}(\log(D_\phi(x)) + \mathbb{E}_q(\log(1 - D_\phi(x))$$

The goal of a GAN is to then tune the distribution q, via a model called the **generator**, such that the discriminator can no longer tell the difference between that

$$\min_q \max_{D_\phi} \mathbb{E}_{\hat{p}}(\log(D_\phi(x)) + \mathbb{E}_q(\log(1 - D_\phi(x))$$

Here, q is the target distribution, p is the distribution we can sample from, and $D_\phi(x)$ is a function providing the probability that a given sample is from the sample distribution, representing a model called the discriminator.

We can use the reparametrization trick from variational autoencoder models to aid in taking the gradient of this learning objective:

$$z \sim q(x)$$

$$\epsilon \sim N(0, z)$$

$$x \sim g(\epsilon)$$

This yields a new expectation term of:

$$\mathbb{E}_\epsilon(\log(1 - D_\phi(x)(x)$$

which is much easier to differentiate.

15.2.2 Training a GAN

The training loop for a GAN is as follows:

1. Fix g, draw sample $\hat{x} \sim g(\epsilon)$ Apply logistic regression to find D_ϕ
2. Fix D_ϕ^* and determine: $\min_g \mathbb{E}_q \left[\log(1 - D_\phi(g(\epsilon)))\right]$, where g is the generator.

15.2.3 Proof of Correctness

Does this actually work?

15.2.3.1 What is the optimal discriminator?

$$L(D_\phi) = \mathbb{E}_p[\log D_\phi(x)] + \mathbb{E}_q[\log(1 - D_\phi(x))]$$

$$\frac{\partial L}{\partial D_\phi} = \hat{p}(x) \cdot \frac{1}{D_\phi(x)} + q(x) \cdot \frac{-1}{1 - D_\phi(x)}$$

Set gradient to 0:

$$\hat{p}(x) - \hat{p}(x) D_\phi^*(x) = q(x) D_\phi^*(x)$$

$$D_\phi^*(x) = \frac{\hat{p}(x)}{\hat{p}(x) + q(x)}$$

Thus, an optimum exists with a given q!

One big question one may ask is if this optimum is a global optimum. While showing this is a little complex, it effectively is convex, unless your generator and discriminator are neural networks.

With the optimal discriminator determined, we turn to the generator.

15.2.3.2 Solve for optimal generator

$$\mathbb{E}_p \left[\log D^*_\phi(x)\right] + \mathbb{E}_q \left[\log(1 - D^*_\phi(x))\right]$$

Here, we substitute the discriminator function we solved for earlier:

$$= \mathbb{E}_p \left[\log \left(\frac{\hat{p}(x)}{\hat{p}(x) + q(x)}\right)\right] + \mathbb{E}_q \left[\log \left(\frac{q(x)}{\hat{p}(x) + q(x)}\right)\right]$$

The above is just our generator loss function! With a little observation, the two terms actually resemble a KL-divergence term, as such:

$$= KL\left(\hat{p} \| \hat{p} + q\right) + KL\left(q \| \hat{p} + q\right)$$

This term reaches an optimum at = q, which intuitively makes sense as the ideal GAN generator, since this is where the generator is producing samples that exactly match the ground truth.

(Note: The final term is known as the Jensen-Shannon Divergence)

15.2.4 How does this connect to EBM?

Consinder the optimization objective for a generalized GAN:

$$\min_q \max_f J(q, f)$$

And that of the Energy-Based Model

$$\max_f \mathbb{E}_p[\log p(x)]$$

Rearranging some terms for the EBM objective function yields:

$$= \max_f \mathbb{E}_p[f(x)] - \hat{\mathbb{E}}_p[\log Z(f)]$$

By definition:

$$Z(f) = \int \exp(f(x))dx$$

This means that the second term reduces to $\log Z(f)$. Making this substitution yields:

$$= \max_f \mathbb{E}_p[f(x)] - \log Z(f)$$

$$= \log \int \frac{\exp(f(x))}{q(x)} q(x)dx$$

$$= \max_q \int q(x) \log \frac{\exp(f(x))}{q(x)} dx$$

$$= \max_q \int q(x) \left[f(x) - \log q(x) \right] dx$$

$$= \max_q \mathbb{E}_q[f(x)] + H(q)$$

$$= \max_f \mathbb{E}_p[f(x)] - \left(\max_q \mathbb{E}_q[f(x)] + H(q) \right)$$

$$= \max_f \min_q \mathbb{E}_p[f(x)] - \mathbb{E}_q[f(x)] - H(q)$$

This is the objective function for a GAN (plus one additional entropy term, called the entropy regularization term).