How does the GAN make $\mathbf{p_{data}} \approx \mathbf{p_{model}}$?

The Generator Loss function we constructed is a proxy to achieve the goal

$$p_{\text{model}} \approx p_{\text{data}}$$

That is: the distribution of samples produced by the Generator is (approximately) the same as the "true" distribution

- we note that we don't know the "true" p_{data}
  - we only have available a sample and those the training set defines an *empirical* distribution

There are several ways to quantify

$$p_{\text{model}} \approx p_{\text{data}}$$

One choice would be the minimization of KL Divergence

- $\mathbf{KL}(p_{\text{data}} || p_{\text{model}})$

As a reminder: we now show that this is equivalent to Maximum Likelihood estimation

Choose p_{model} to Minimize

$$\begin{array}{lll}
\mathbf{KL}(p_{\text{data}} || p_{\text{model}}) & = & \int_{\mathbf{x}} p_{\text{data}}(\mathbf{x}) \left(\log \frac{p_{\text{data}}(\mathbf{x})}{p_{\text{model}}\mathbf{x})} \right) d\mathbf{x} & \text{Definition of KI} \\
& = & \mathbf{E}_{\mathbf{x} \in p_{\text{data}}} \log(p_{\text{data}}(\mathbf{x}) - \log(p_{\text{model}}(\mathbf{x})) & \text{Definiton of log} \\
\text{minimizing KL} \\
& \approx & \mathbf{E}_{\mathbf{x} \in p_{\text{data}}} - \log(p_{\text{model}}(\mathbf{x})) & \text{Since \$} \log(p_{\text{data}}
\end{array}$$

So minimizing $\mathbf{KL}$ is equivalent to minimizing the Negative Log Likelihood. mm

Notice that the expectation is over the "true" distribution p_{data}.

The expectation is certainly reasonable for training put perhaps not best for the purposes of generating synthetic data

- Measures fidelity to training data
- NOT how "realistic" the synthetic data is
- the penalty for p_{model} placing large probability mass around a particular $\hat{\mathbf{x}}'$ is small when $p_{\text{model}} \hat{\mathbf{x}}' \approx 0$
    - so Generator may create large quantity of synthetic data that is improbable given the training set

If we knew the true p_{data}, a better objective to minimize for the purpose of generating synthetic data would be the similar

$$\mathbf{KL}(p_{\text{model}}||p_{\text{data}})$$

which is equivalent to maximizing

$$\mathbf{E}_{\mathbf{x}\in p_{\text{model}}} - \log(p_{\text{data}}(\mathbf{x}))$$

The expectation is over the synthetic data, not the true data

- $\log(p_{\text{data}}(\mathbf{x}))$ is defined as log of Perplexity
    - an element of "surprise" in seeing $\mathbf{x}$
- So the expectation asks: for each synthetic datum generated, how likely is it to occur in the true distribution ?

This is merely a theoretical argument

- In practical terms: we only have empirical p_{data}
- So can't evaluate log Perplexity $p_{\mathrm{data}}(\hat{\mathbf{x}})$ for $\hat{\mathbf{x}} \in p_{\mathrm{model}}$ unless synthetic $\hat{\mathbf{x}}$ replicates a sample in the training data

Jensen-Shannon Divergence

We have observed that the KL divergence is *not* symmetric

$$\mathbf{KL}(P||Q) \neq \mathbf{KL}(Q||P)$$

because the expectations are taken over different distributions.

An alternative measure of similarity of two distributions is the Jensen-Shannon Divergence (JSD)

$$
\begin{aligned}
\mathrm{JSD}(P||Q) &= \mathrm{JSD}(Q||P) \\
&= \frac{1}{2}\,\mathrm{KL}\left(P\,||\,\frac{P+Q}{2}\right) + \\
&\quad \frac{1}{2}\,\mathrm{KL}\left(Q\,||\,\frac{P+Q}{2}\right)
\end{aligned}
$$

This measure is

- symmetric
- is a kind of mixture of $\mathbf{KL}(P||Q)$ and $\mathbf{KL}(Q||P)$.

[Huszar (https://arxiv.org/pdf/1511.05101.pdf)](https://arxiv.org/pdf/1511.05101.pdf) has a Generalized JSD which interpolates between the two terms

$$\begin{aligned} \mathrm{JSD}_\pi(P||Q) &= \mathrm{JSD}(Q||P) \\ &= \pi\, \mathrm{KL}\left(P \,||\, \pi P + (1-\pi)Q\right) + \\ &\quad (1-\pi)\, \mathrm{KL}\left(Q \,||\, \pi P + (1-\pi)Q\right) \end{aligned}$$

The Generalized JSD

- **Not** symmetric although
  $$\mathrm{JSD}_\pi(P||Q) = \mathrm{JSD}_{1-\pi}(Q||P)$$
- Is similar to Maximum Likelihood when $\pi \approx \epsilon$
- Is similar to $\mathbf{KL}(Q||P)$ when $\pi \approx (1-\epsilon)$
  $$\frac{\mathrm{JSD}_{1-\epsilon}(P||Q)}{1-\epsilon} \approx \mathrm{KL}(Q||P)$$

In implementing Generalized JSD

- The Discriminator is trained (as usual) on a mix of real an fake examples
    - But *not* in equal numbers
    - π is fraction of samples from Q
    - $(1 - \pi)$ is fraction of samples from P
    - $\pi < \frac{1}{2}$: real samples over represented
    - $\pi > \frac{1}{2}$: biased toward Q
- Explains why we often see training with Generator updated twice for each update of Discriminator ?

Adversarial Training and the Jensen-Shannon Divergence

The Discriminator Loss $\mathcal{L}_D$, summed over all examples (ignoring the $\frac{1}{2}$ from the previous presentation where we assumed equal number of Real and Fake)

$$\mathcal{L}_D \quad = \quad - \left(\mathbf{E}_{\mathbf{x}^{(\mathbf{i})} \in p_{\text{data}}} \log D(\mathbf{x}^{(\mathbf{i})}) + \mathbf{E}_{\mathbf{x}^{(\mathbf{i})} \in p_{\text{model}}} \log\left(1 - D(\mathbf{x}^{(\mathbf{i})})\right) \right) \quad D(G(\mathbf{z})) = \mathbf{x}$$

We also showed that the optimal Discriminator results in

$$D^*(\mathbf{x}) = \frac{p_{\text{data}}(\mathbf{x})}{p_{\text{model}}(\mathbf{x}) + p_{\text{data}}(\mathbf{x})}$$

Plugging $D^*(\mathbf{x})$ into $\mathcal{L}_D$ (Goodfellow Equation):

$$\begin{aligned}
\mathcal{L}_D &= -\left(\mathbf{E}_{\mathbf{x^{(i)}} \in p_{\text{data}}} \log \frac{p_{\text{data}}(\mathbf{x})}{p_{\text{model}}(\mathbf{x})+p_{\text{data}}(\mathbf{x})} + \mathbf{E}_{\mathbf{x^{(i)}} \in p_{\text{model}}} \log \frac{p_{\text{model}}(\mathbf{x})}{p_{\text{model}}(\mathbf{x})+p_{\text{data}}(\mathbf{x})}\right) \\
&= -(\mathbf{KL}(p_{\text{data}}||p_{\text{model}}(\mathbf{x}) + p_{\text{data}}(\mathbf{x})) + \mathbf{KL}(p_{\text{model}}||p_{\text{model}}(\mathbf{x}) + p_{\text{data}}(\mathbf{x}) \\
&= -\left(\log 4 + \mathbf{KL}(p_{\text{data}}||\frac{p_{\text{model}}(\mathbf{x})+p_{\text{data}}(\mathbf{x})}{2}) + \mathbf{KL}(p_{\text{model}}||\frac{p_{\text{model}}(\mathbf{x})+p_{\text{data}}(\mathbf{x})}{2}\right) \\
&= -(\log 4 + 2 * \text{JSD}(p_{\text{data}}||p_{\text{model}}))
\end{aligned}$$

Thus Goodfellow proves that solving the minimax optimally minimizes the JSD divergence between p_{data} and p_{model}.

To summarize

- $\mathcal{L}_D$ is implemented by KL Divergence
- *Under the assumption* that the Discriminator can train to be the **optimal** adversary
    - $\mathcal{L}_D$ becomes equivalent to the Jensen-Shannon Distance