How does the GAN make $\backslash\mathbf{pdata} \approx \backslash\mathbf{pmodel}$?

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

$$\backslash\mathrm{pmodel} \approx \backslash\mathrm{pdata}$$

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" $\backslash\mathrm{pdata}$
    - we only have available a sample and those the training set defines an *empirical* distribution

There are several ways to quantify

$$\pmodel \approx \pdata$$

One choice would be the minimization of KL Divergence

- $\KL(\pdata || \pmodel)$

An alternative, still using KL Divergence

- $\KL(\pmodel || \pdata)$

Which is a better choice ?

In order to answer the question, we begin with a few preliminaries

- Definition of KL Divergence
- Proving
    - minimizing KL Divergence increases log-likelihood

Definition of KL Divergence

As a reminder of the definition of KL Divergence

$$
\begin{aligned}
\mathrm{KL}(p||q) &= -\sum_x p(x) \log q(x) + \sum_x p(x) \log p(x) \\
&= \sum_x p(x) * (\log p(x) - \log q(x)) \\
&= \backslash\mathrm{E}_{\backslash\mathbf{x} \sim p} (\log p(x) - \log q(x))
\end{aligned}
$$

You can see that it is

- the point-wise difference between the (log) probability of $\backslash\mathbf{x}$ in distributions p and q
- averaged over the distribution of $\backslash\mathbf{x} \sim p$

and thus is a point-wise measure of the dis-similarity of the two distributions.

We note that the KL Divergence is *not symmetric*

$$\backslash\mathrm{KL}(\backslash\mathrm{pdata}||\backslash\mathrm{pmodel}) \neq \backslash\mathrm{KL}(\backslash\mathrm{pmodel}||\backslash\mathrm{pdata})$$

so the two choices are different.

- both are expectations
- but over *different* distributions

KL Divergence leads to Maximum Likelihood Estimation

We now show that using $\backslash \mathrm{KL}(\backslash \mathrm{pdata}||\backslash \mathrm{pmodel})$ as a loss function

- results in a estimation of the model distribution $\backslash \mathrm{pmodel}$
- that is the Maximum Likelihood estimator of the training examples (represented by $\backslash \mathrm{pdata}$)

That is

- $\backslash \mathrm{pmodel}$ is the best explanation of the training dataset $\backslash \mathrm{pdata}$

Choosing $\backslash\mathrm{pmodel}$ to Minimize gives

$$
\begin{aligned}
\backslash\mathrm{KL}(\backslash\mathrm{pdata}||\backslash\mathrm{pmodel}) &= \int_{\backslash\mathrm{x}} \backslash\mathrm{pdata}(\backslash\mathrm{x}) \left(\log \frac{\backslash\mathrm{pdata}(\backslash\mathrm{x})}{\backslash\mathrm{pmodel}\backslash\mathrm{x})}\right) d\backslash\mathrm{x} \\
&= \backslash\mathrm{E}_{\backslash\mathrm{x}\in\backslash\mathrm{pdata}} \log(\backslash\mathrm{pdata}(\backslash\mathrm{x})) - \log(\backslash\mathrm{pmodel}(\backslash\mathrm{x})) \\
\text{minimizing KL} & \\
&\approx \backslash\mathrm{E}_{\backslash\mathrm{x}\in\backslash\mathrm{pdata}} - \log(\backslash\mathrm{pmodel}(\backslash\mathrm{x}))
\end{aligned}
$$

So minimizing $\backslash\mathrm{KL}$ is equivalent to

- minimizing the Negative Log Likelihood
- in other words: *maximizing* the Log Likelihood

Choosing the KL Divergence

The first choice

$$\backslash\mathrm{KL}(\backslash\mathrm{pdata}||\backslash\mathrm{pmodel}) = \backslash\mathrm{E}_{\backslash\mathrm{x}\sim\backslash\mathrm{pdata}}(\log \backslash\mathrm{pdata}(x) - \log \backslash\mathrm{pmodel}(x))$$

maximizes $\log(\backslash\mathrm{pmodel}(\backslash\mathrm{x}))$ for $\backslash\mathrm{x} \in \backslash\mathrm{pdata}$

- $\backslash\mathrm{pmodel}$ assigns high probability to Real examples
- model creates Real examples with high probability

By way of analogy with measures for Classification

- the expectation over $\backslash\mathrm{pdata}$ emphasizes Recall over Precision

We can achieve high Recall

- by reducing chance of False Negatives (FN)
- even if it increases chance of False Positives (FP)

In the GAN context this means

reducing FN $\rightsquigarrow$ \pmodel assigns high probability to each training example in

increasing FP $\rightsquigarrow$ \pmodel assigns high probability to \x $\notin$ \pdata

The second choice $\backslash\mathrm{KL}(\backslash\mathrm{pmodel}||\backslash\mathrm{pdata})$

$$\backslash\mathrm{KL}(\backslash\mathrm{pmodel}||\backslash\mathrm{pdata}) = \backslash\mathrm{E}_{\backslash\mathrm{x}\sim\backslash\mathrm{pmodel}}(\log \backslash\mathrm{pmodel}(x) - \log \backslash\mathrm{pdata}(x))$$

maximizes $\backslash\mathrm{pdata}(x)$ for $\backslash\mathrm{x} \in \backslash\mathrm{pmodel}$

- emphasizes that synthetic examples are "realistic"
    - highly probable, as defined by the empirical distribution (training data) $\backslash\mathrm{pdata}$

This ("realistic examples") might be the more desirable property than "high fidelity" to the training data.

Continuing with our Recall versus Precision analogy, this measure

- increases Precision by reducing False Positives
    - examples generated by \pmodel are likely according to \pdata

So it seems as if the second choice $\KL(\pmodel||\pdata)$ may be more desirable.

But we don't know the true $\pdata$!

- we only have an empirical sample: the training dataset
- so, in practical terms: we can't maximize it

Thus, practical considerations lead us to the first choice.

Jensen-Shannon Divergence

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

$$\backslash\mathrm{KL}(P||Q) \neq \backslash\mathrm{KL}(Q||P)$$

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 $\backslash\mathrm{KL}(P||Q)$ and $\backslash\mathrm{KL}(Q||P)$.

[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

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

The Generalized JSD

- **Not** symmetric although $\mathrm{JSD}_\pi(P||Q) = \mathrm{JSD}_{1-\pi}(Q||P)$

In [ ]:

Huszar shows that, for small values of π

$$\frac{\mathrm{JSD}_{\pi}(P||Q)}{\pi} \approx \mathrm{KL}\left(\, P \,||\, Q\right)$$

and

$$\frac{\mathrm{JSD}_{1-\pi}(P||Q)}{1-\pi} \approx \mathrm{KL}\left(\, Q \,||\, P\right)$$

In the first case

- $\mathrm{JSD}_{\pi}(P||Q)$ is proportional to Maximum Likelihood

In the second case

- $\mathrm{JSD}_{1-\pi}(P||Q)$ is proportional to $\mathrm{KL}\left(\, Q \,||\, P\right)$

In implementing Generalized JSD

- The Discriminator is trained (as usual) on a mix of real and 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 $\loss_D$

- summed over all examples
  - (ignoring the $\frac{1}{2}$ from the previous presentation where we assumed equal number of Real and Fake)

is

$$\loss_D \quad = \quad - \left(\E_{\x^\ip \in \pdata} \log D(\x^\ip) + \E_{\x^\ip \in \pmodel} \log \left(1 - D(\x^\ip) \right) \right)$$

We also showed that the optimal Discriminator results in

$$D^*(\x) = \frac{\pdata(\x)}{\pmodel(\x) + \pdata(\x)}$$

```
In [ ]:
```

Plugging $D^*(\backslash\mathrm{x})$ into $\backslash\mathrm{loss}_D$ (Goodfellow Equation):

$$
\begin{aligned}
\backslash\mathrm{loss}_D &= -\Big(\backslash\mathrm{E}_{\backslash\mathrm{x}^{\backslash\mathrm{ip}}\in\backslash\mathrm{pdata}}\log\frac{\backslash\mathrm{pdata}(\backslash\mathrm{x})}{\backslash\mathrm{pmodel}(\backslash\mathrm{x})+\backslash\mathrm{pdata}(\backslash\mathrm{x})} + \backslash\mathrm{E}_{\backslash\mathrm{x}^{\backslash\mathrm{ip}}\in\backslash\mathrm{pmodel}}\log\frac{\backslash\mathrm{p}}{\backslash\mathrm{pmodel}(} \\
&= -(\backslash\mathrm{KL}(\backslash\mathrm{pdata}||\backslash\mathrm{pmodel}(\backslash\mathrm{x}) + \backslash\mathrm{pdata}(\backslash\mathrm{x})) \\
&\quad + \backslash\mathrm{KL}(\backslash\mathrm{pmodel}||\backslash\mathrm{pmodel}(\backslash\mathrm{x}) + \backslash\mathrm{pdata}(\backslash\mathrm{x})) \\
&= -\Big(\log 4 + \backslash\mathrm{KL}(\backslash\mathrm{pdata}||\frac{\backslash\mathrm{pmodel}(\backslash\mathrm{x})+\backslash\mathrm{pdata}(\backslash\mathrm{x})}{2}) \\
&\quad + \backslash\mathrm{KL}(\backslash\mathrm{pmodel}||\frac{\backslash\mathrm{pmodel}(\backslash\mathrm{x})+\backslash\mathrm{pdata}(\backslash\mathrm{x})}{2}\Big) \\
&= -(\log 4 + 2 * \mathrm{JSD}(\backslash\mathrm{pdata}||\backslash\mathrm{pmodel}))
\end{aligned}
$$

The above equations shows that

- minimizing KL Divergence (second line above)
- under the assumption that the Discriminator can train to be the **optimal** adversary

results in $\backslash\mathrm{loss}_D$ becoming equivalent to Jensen-Shannon Distance (last line above)

So solving the minimax optimally minimizes the JSD divergence between $\backslash\mathrm{pdata}$ and $\backslash\mathrm{pmodel}$.

In [2]:

```
print("Done")
```

```
Done
```