In [1]:
```
%run Latex_macros.ipynb
%run beautify_plots.py
```

Wasserstein GAN: Motivation

paper (https://arxiv.org/pdf/1701.07875.pdf)

To summarize what we have learned about standard GANs:

- Adversarial Training minimizes the Jensen Shannon Distance between p_{model} and p_{data}
- They have the reputation for being difficult to train
    - A Discriminator that is too good, too soon inhibits the ability of the Generator to learn to generate
    - The Generator may "mode collapse" and not produce a variety of outputs

The *Wasserstein GAN (WGAN)* is a pair of Neural Networks NN:

- the Generator
- the Discriminator
    - technically, this is a "critic"
        - rather than producing a probability of "Real"
        - it produces a "score" measuring how real the input is
            - larger negative: more real
            - larger positive: less real

The pair is trained to minimize an *approximation* of

$$\mathbb{W}(p_{\text{data}}, p_{\text{model}})$$

where $\mathbb{W}$ is the *Wasserstein Distance*, also know as the *Earth Move Distance (EMD)* measure.

Earth Move Distance (EMD)

Aside

You need some knowledge of Measure Theory to understand the math.

In the absence, there are two good blogs I recommend in order to get a flavor

- Sorta Insightful (https://www.alexirpan.com/2017/02/22/wasserstein-gan.html)
- Wen (https://arxiv.org/pdf/1904.08994.pdf)

Like the KL and Jensen-Shannon Distances, the EMD is a measure of the difference between two distributions. p_{data} and p_{model}.

It has an intuitive explanation

> *The minimum amount of "work" involved in moving probability mass between the two distributions in order to make them identical*

"Work" means: the product of

- the quantity $\gamma(x, y)$ of the mass moved from x to y
- and the distance $||x - y||$ it is moved

We can easily illustrate with two discrete distributions (example from Wen (https://arxiv.org/pdf/1904.08994.pdf)

Let P, Q be the two distributions, represented as vectors since there are discrete and measured over the same indices.

$$\begin{aligned} P &= [3, 2, 1, 4] \\ Q &= [1, 2, 4, 3] \end{aligned}$$

P_i (resp., Q_i) is the probability as i in each of the distributions, for $1 \leq i \leq 4$.

For illustration, we will move

- a quantity δ_i of probability in P between adjacent indices $(i-1)$ and i
- in order to make $P_i = Q_i$ (Q_i remains fixed)
- The distance is 1 (and hence work is equal to quantity moved).

We can define δ_i recursively:

$$\begin{aligned} \delta_0 &= 0 \\ \delta_{i+1} &= \delta_i + P_i - Q_i \end{aligned}$$

That is, the amount δ_{i+1} moved from P_i in order to make $P_i = Q_i$ is

- the difference $(P_i - Q_i)$ between original value of P_i and Q_i
- plus the additional quantity δ_i that was moved into P_i

$$
\begin{aligned}
\delta_0 &= 0 \\
\delta_1 &= 0 + 3 - 1 = 2 \\
\delta_2 &= 2 + 2 - 2 = 2 \\
\delta_3 &= 2 + 1 - 4 = -1 \\
\delta_4 &= -1 + 4 - 3 = 0
\end{aligned}
$$

Work is positive so taking absolute values

$$\mathbb{W}(P, Q) = \sum_{i=1}^{4} 1 * |\delta_i| = 5$$

For continuous distributions

$$\mathbb{W}(p_{\text{data}}, p_{\text{model}}) = \inf_{\gamma \in \Pi(p_{\text{data}}, p_{\text{model}})} \mathbb{E}_{(x,y)\sim\gamma} \|x - y\|$$

where

- $\Pi(p_{\text{data}}, p_{\text{model}})$ are the set of possible joint distributions with marginal p_{data} and p_{model}
- γ is a quantity to move from x to y (for all x, y)
    - distance between x and y is $\|x - y\|$
- $\inf$ is the infimum (Greatest Lower Bound)

Approximation of $\mathbb{W}(p_{\text{data}}, p_{\text{model}})$

Warning: the math is stated without much explanation

The infimum is intractable (or at least: not practical to compute).

Equation 2 in the paper states that for certain functions f, the distance is also equal to

$$\mathbb{W}(p_{\text{data}}, p_{\text{model}}) = \inf_{\|f\|_2 \leq 1} \mathbb{E}_{x \sim p_{\text{data}}} f(x) \; - \; \mathbb{E}_{x \sim p_{\text{model}}} f(x)$$

One can look at f as a "score" of x being "Real" (not fake) where

- a high negative score is a highly confident "Real"
- a high positive score is a highly confident "Fake"

The goal is

- for function f to create a *large spread* between scores of Real and Fake.
- for function f to be *approximated* by the Discriminator D_Θ with weights Θ_D

Under certain conditions on f, finding $\mathbb{W}$ is equivalent to solving

$$\max_{\Theta_D \in \mathcal{W}} \mathbb{E}_{x \sim p_{\text{data}}} D_{\Theta_D}(x) \; - \; \mathbb{E}_{x \sim p_{\text{model}}} D_{\Theta_D}(x)$$

where $\mathcal{W}$ is a "compact" space of possible weights

Since the Discriminator no longer produces binary categorical values, it is more appropriate to call it a *Critic*.

That is: we solve for Critic weights such that the scores it produces have a large spread between Real and Fake.

But: what does this mean ?

For those (like me) struggling with the math, here are the implications from a practical perspective

- Scores for true Real is negative, for Fake is positive
- $\mathcal{L}_G$ will implement: minimize (make most negative) the score assigned to Fakes
- $\mathcal{L}_D$ will implement: "maximize the spread of scores between Real and Fake"
    - by minimizing the sum of
        - sum of scores for Real examples
        - minus sum of scores for Fake examples (i.e., Discriminator goal is for Fakes to have positive scores)
- "Compact" Θ_D will be achieved by clipping
    - restricting elements of Θ_D to a small numerical range
    - by clipping the weights after a gradient update step for D

- $\mathcal{L}_D$ will dispense with the $\log$ since the Discriminator produces scores rather than probabilities
    - we see terms $D(\mathbf{x^{(i)}})$ and $1 - D(\mathbf{x^{(i)}})$
    - rather than $\log D(\mathbf{x^{(i)}})$ and $1 - \log D(\mathbf{x^{(i)}})$

When we visit the code, we will see these elements in practice.

Did I really need to change to EMD ?

The Wasserstein GAN avoids many of the problems associated with the plain GAN.

To some extent, this is due to replacing the Discriminator with a Critic

- unbounded scores in the WGAN versus bounded probabilities in the plain GAN

- There are mathematical problems with Expectation Maximization (KL distance) and Jensen-Shannon (JS)distance
    - the terms $\log(p_{\text{model}}(\mathbf{x}))$ and $\log(p_{\text{data}}(\mathbf{x}))$ appear
    - if p_{data} and p_{model} don't completely overlap (a possibility especially early in training)
        - we take logs of 0, which is infinite (negative)
    - No such problem with EMD

- No vanishing gradients with EMD
    - with KL and JS distance the true derivative goes to 0
    - no such problem with EMD
        - The Critic's scores are not bounded, so can't saturate
    - Thus: we can train the Discriminator to convergence immediately
        - No danger of being too good too soon
        - When we see code we will observe
            - The number of steps of Discriminator update is a multiple of the number of steps of Generator update
- No mode collapse with EMD
    - with a fixed Discriminator (classifier), the Generator in a plain GAN will seek out examples with highest probability of being mis-classified as Real

Code