5 Minute Summary

1. **The Goal:**
    - Imagine you have a bunch of data points, each with many features (like pixels in an image or characteristics of a song).
    - The goal of t-SNE is to take these high-dimensional data points and find a way to represent them in a lower-dimensional space (usually 2D or 3D). This makes it easier to visualize and understand the relationships between the points.
2. **Local Relationships:**
    - t-SNE focuses on preserving the relationships between nearby points. If two points are close together in the high-dimensional space, t-SNE wants to keep them close together in the lower-dimensional space too.
3. **Similarity:**
    - It measures the similarity between points using a Gaussian (bell-shaped) curve. If two points are very similar, the curve is tall. If they're not very similar, the curve is short.
    - This similarity is based on distances between points: the closer two points are, the more similar they're considered to be.
4. **Finding a Good Fit:**
    - t-SNE tries different arrangements of the points in the lower-dimensional space until it finds one that preserves the similarities as well as possible.
    - It adjusts the positions of the points in the lower-dimensional space to minimize the difference between the similarities of the original high-dimensional data and the lower-dimensional representation.
5. **Iterative Process:**
    - t-SNE keeps tweaking the positions of the points until it finds a good fit. It's like solving a puzzle: you move the pieces around until everything fits together nicely.
6. **Visualizing the Result:**

- Once t-SNE has found a good fit, you can plot the points in the lower-dimensional space. Because it preserves the local relationships well, you'll often see clusters of points that are similar to each other.
- This visualization helps you understand the structure of your data and identify patterns that may not have been obvious in the original high-dimensional space.

In essence, t-SNE is like a translator for your data. It takes the complex language of high-dimensional space and translates it into a simpler, more understandable form that you can easily interpret and explore.

Mathematical Explanation

1. **Compute Pairwise Similarities:**
   - Given a dataset $X = \{x_1, x_2, ..., x_N\}$ consisting of N high-dimensional data points, t-SNE first computes pairwise similarities between all data points. It uses a Gaussian kernel function to measure the similarity between two points x_i and x_j:

$$p_{j|i} = \frac{\exp(-||x_i - x_j||^2/(2\sigma_i^2))}{\sum_{k \neq i} \exp(-||x_i - x_k||^2/(2\sigma_i^2))}$$

   - Here, $p_{j|i}$ represents the conditional probability of choosing data point x_j given x_i, and σ_i is the variance of the Gaussian kernel centered at x_i. The choice of σ_i influences the scale of the local neighborhoods.
2. **Construct Similarity Distributions:**
   - Based on the computed pairwise similarities, t-SNE constructs two probability distributions:
     - $P = \{p_{j|i}\}$: The high-dimensional similarity distribution. It captures the relationships between data points in the original high-dimensional space.
     - $Q = \{q_{ji}\}$: The low-dimensional similarity distribution. It represents the relationships between points in a lower-dimensional space (e.g., 2D or 3D).
3. **Optimize Embedding Positions:**
   - t-SNE aims to find an embedding of the data points in the lower-dimensional space such that the low-dimensional similarity distribution Q closely matches the high-dimensional similarity distribution P. It achieves this by minimizing the KL divergence between the two distributions:

$$\mathrm{KL}(P||Q) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{ji}}$$

   - Here, $p_{j|i}$ and q_{ji} represent the conditional probabilities in P and Q, respectively.

4. **Gradient Descent Optimization:**
    - t-SNE uses gradient descent optimization to minimize the KL divergence. It iteratively adjusts the positions of the embedded points $Y = \{y_1, y_2, ..., y_N\}$ in the lower-dimensional space to reduce the discrepancy between P and Q.
    - The gradient of the cost function with respect to the embedded points Y is computed, and the points are updated accordingly:

$\frac{\partial C}{\partial y_i} = 4\sum_j (p_{j|i} - q_{ji})(y_i - y_j)(1 + \|y_i - y_j\|^2)^{-1}$

    - This update rule adjusts the positions of the embedded points based on the difference between the high-dimensional and low-dimensional similarity distributions.
5. **Convergence:**
    - The optimization process continues until convergence, meaning that the positions of the embedded points stabilize and the KL divergence between P and Q is minimized as much as possible.
6. **Final Embedding:**
    - Once the optimization process is complete, the resulting embedding of the data points in the lower-dimensional space Y captures the local structure of the data observed in the original high-dimensional space.

In summary, t-SNE iteratively optimizes the embedding of high-dimensional data points in a lower-dimensional space by minimizing the KL divergence between high-dimensional and low-dimensional similarity distributions. This results in a visually meaningful representation of the data that preserves local relationships well.