Goal

Problem statement:

> *Given a point cloud* $X = \{x_i : x_i \in \mathbb{R}^3, i \in [N]\}$, *predict a function* $\mathcal{O} : \mathbb{R}^3 \to [0, 1]$, *which represents the volume represented by the point cloud.*

- Talk: Using Irregular Latent Grid to solve the problem.
- Presenter: Annada Behera

Prerequisites

1. Farthest point sampling
2. k-nearest neighbor
3. PointNet
4. Vector quantization
5. Nadaraya-Watson estimator

Farthest point sampling

1. Pick a point p from the point cloud, X arbitrarily and add it to the sampled points, $X_M = \{p\}$.
2. Find the point, p' in the point cloud X such that the distance between p and p' is maximum.
$$p' = \underset{q \in X \setminus \{p\}}{\arg\max} \|p - q\|_2 \tag{1}$$
This takes $O(N)$ time.
3. Add it to the sampled point, $X_M \leftarrow X_M \cap \{p'\}$.
4. Repeat step (2) and (3), until M number of points are sampled, taking $O(M)$ time.

Total time: $O(MN)$.

k-nearest neighbor

Given a point $p \in X$ in the point cloud, the k-nearest neighbor can be computed as,

1. Find the distance between p and all other point, as a ordered set S,

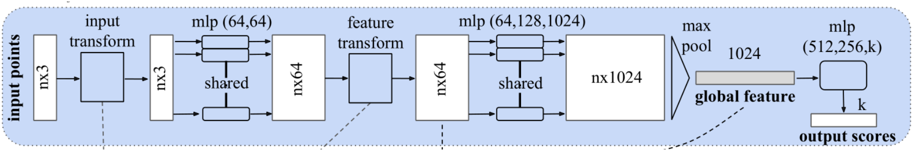
$$S = \{\|q - p\|_2 : q \in X \setminus \{p\}\} \tag{2}$$

   Takes $O(N)$ time.

2. Sort the ordered set, S in $O(N \log N)$ time.
3. Return the first k elememts, in $O(k)$.

Since $k < M < N$, total time: $O(N \log N)$.

PointNet embedding vector



For every point p in X, the embedding in $\mathbb{R}^C$ latent space is given as a neural network, defined as a series of computations as follows,

$$e_i = \textsc{MaxPool}(t_{n\times C}) \tag{3}$$

$$t_{n\times C} = W^{(0)}_{C\times 128} W^{(1)}_{128\times 64} s_{n\times 64} \tag{4}$$

$$s_{n\times 64} = W^{(2)}_{64\times 64} r_{n\times 64} \odot r_{n\times 64} \tag{5}$$

$$r_{n\times 64} = W^{(3)}_{64\times 64} q_{n\times 3} \tag{6}$$

$$q_{n\times 64} = W^{(4)}_{3\times 3} p_{n\times 3} \odot p_{n\times 3} \tag{7}$$

Vector quantization

For a given vector, $z \in \{z_i \in \mathbb{R}^n, i \in [M]\}$, *vector quantization* is the process of moving the vector to another vector which is closest to it from a given dictionary D,

$$\forall i \in M, z_i = \underset{\hat{z}_i \in D}{\arg\max} \|\hat{z}_i - z_i\| \tag{8}$$

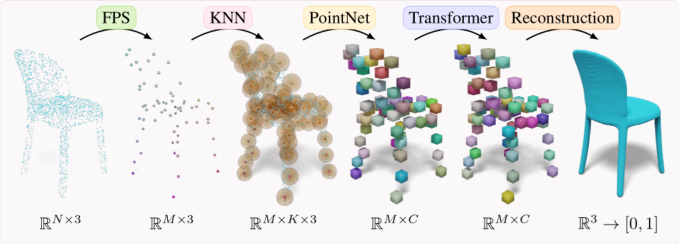
Nadaraya-Watson estimator

The Nadaraya-Watson estimator of a latent vector z_x for it's corresponding point $x \in \{x_i \in \mathbb{R}^3, i \in [M]\}$ is given as,

$$z_x = NW(x, z) = \frac{\sum_{i \in [M]} z_i \exp(-\beta \|x - x_i\|^2)}{\sum_{i \in [M]} \exp(-\beta \|x - x_i\|^2)} \tag{9}$$

where, β is the smoothness of interpolation.

Shape Reconstruction



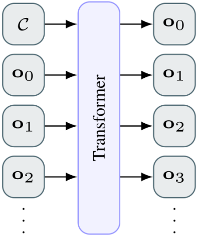
The loss is a weight loss between binary-cross entropy loss and the commitment loss,

$$L = \mathbb{E}_{x\in\mathbb{R}^3} \mathrm{B\scriptstyle{CE}}(\mathcal{O}, \mathcal{O}') + \lambda \mathbb{E}_{x\in\mathbb{R}^3} [E_{x\in[M]} \| \hat{z}_i^{(l)} - z_i^{(l)} \|^2] \tag{10}$$

Autoregressive Generative Modeling

1. Quantize the latents,
$$z_i \in \{0, 1, \cdots, D\} \tag{11}$$
2. Quantize the points,
$$x_i \rightarrow (x_{i,1}, x_{i,2}, x_{i,3}) \tag{12}$$
where $(\cdot_{(\cdot,i)}) \in [255]$.
3. Create a sequence, $\mathcal{S}$, an ordered set,
$$\mathcal{S} = \{(x_{0,1}, x_{0,2}, x_{0,3}, z_0), \tag{13}$$
$$(x_{1,1}, x_{1,1}, x_{1,1}, z_1), \cdots, \tag{14}$$
$$(x_{(M-1,1),}, x_{(M-1,2)}, x_{(M-1,3),z_{M-1})}\} \tag{15}$$
4. The final model is,
$$\Pr[\mathcal{S}|\mathcal{C}] = \prod_{i=0}^{M-1} \Pr[o_i|o_{<i}, \mathcal{C}] \tag{16}$$

AGM(cond.)



$\mathcal{C}$
$\mathbf{o}_0$
$\mathbf{o}_1$
$\mathbf{o}_2$
Transformer
$\mathbf{o}_0$
$\mathbf{o}_1$
$\mathbf{o}_2$
$\mathbf{o}_3$