VoteNet architecture

1. The entire network can be splitted into 2 parts :
   - process existing points to generate votes
   - use the votes to propose & classify objects

Vote generation

1. Given N point clouds, they generate M votes.
2. Each vote has a 3D coordinate and a high dimensional vector.
3. There are 2 major steps :
   - point cloud feature learning through a backbone network.
   - learned Hough voting from seed points.

Point Cloud Feature Learning

1. PointNet++ is used as the backbone
2. The backbone network outputs a subset of the input points with XYZ and a C-dimensional feature vector.
3. The results are M seed points of dimension $(3+C)$.
4. Each seed corresponds to one vote.

Hough voting

1. Given a set of seed point $\{s_i\}_{i=1}^{M}$ where $s_i = [x_i ; f_i]$. $x_i \in R^3$. $f_i \in R^C$ a voting module generates votes from each seed independently.

2. The voting module is a MLP, Relu w BN.

3. The MLP takes seed feature f_i

outputs . Euclidean space offset $\Delta x_i \in R^3$

. feature off-set $\Delta f_i \in R^C$.

4. The vote $v_i = [y_i ; g_i]$ generated from the seed s_i.

$y_i = x_i + \Delta x_i$, $g_i = f_i + \Delta f_i$.

5. The predicted offset Δx_i is explicitly supervised by a regression loss :

$$L_{vote\text{-}reg} = \frac{1}{M_{pos}} \sum_i \| \Delta x_i - \Delta x_i^* \| \mathbb{1}_{s_i \text{ on obj}}.$$

count of total no. of seed on obj surface

indicates whether a seed point s_i is an on obj. surface.

GT displacement from the seed position x_i to the bounding box center of the obj it belongs to.

6. An essential difference b/t votes & seeds : votes generated from seeds on the same obj are now closer to each other than the seeds are.

7. This is referred to as semantic - aware locality.

Object Proposal & Classification from Votes

1. Given the votes, they:
   - cluster them
   - use the clusters to generate obj. proposal & classify them.

Clustering

1. is done through uniform sampling & spatial proximity.
2. From a set of votes $\{ v_i = [y_i ; g_i] \in R^{3+C} \}_{i=1}^{M}$:
   - sample a subset of K points with farthest point sampling based on $\{ y_i \}$
   - get $\{ v_{i_k} \}_{k=1}^{K}$
3. Form K clusters by finding neighboring votes to each of the v_{i_k}'s 3D location:

$$C_k = \{ v_i^{(k)} \mid \| v_i - v_{i_k} \| \leq r \} \quad \forall k = 1, \ldots, K.$$

Obj. proposal and classification

1. Given a vote cluster $C = \{ w_i \}_{i=1}^{n}$ where $w_i \in [z_i ; h_i]$
   - $z_i \in R^3$: vote location
   - $h_i \in R^C$: vote feature
2. Locally normalized the vote location: $z_i' = (z_i - z_j)/r$.
3. Obj proposal is generated by passing C through a PointNet-like module:

$$p(C) = MLP_2 \{ \max_{i=1,\ldots,n} \{ MLP_1 ([z_i' ; h_i]) \} \}$$

max-pooled channel wise

4. The proposal p is a vector with an objness score, bounding box parameters (center, heading, scale) and semantic classification scores.

loss function

1. The loss fun in the proposal & classification stage consists of :
   - objectness
   - bounding box regression
   - semantic classification