Weekly Update

Hanchen Wang

Ph.D Candidate in
Engineering, University of Cambridge

September 30, 2019

Overview

1. project I - Points Cloud Sampler
   - progress. 1: snippets on the points cloud dataset and tasks
   - progress. 2: PointNet retrospectives: set functions
   - progress. 3: shape retrieval with deep metric learning

2. project II - Latent Graph Neural Nets
   - progress: add latent GNN module onto the backbone
   - next step: combine with SkipNet, figure out details in design

Performance of Trained Model with Different Set Functions

Set Function	# of points	Training Accuracy	Test Accuracy	Comment
maxpool	512	94.8%	87.8%	
maxpool	1024	96.1%	85.7%	default setting
maxpool	2048	96.9%	88.4%	
avgpool	512	95.9%	16.8%	
avgpool	1024	96.4%	26.9%	
avgpool	2048	95.2%	9.6%	
DeepSet[1]	100	N/A	82%	
DeepSet	5000	N/A	90.3%	

Table: benchmark on ModelNet40 Classification Task

- for object classification, output should be unchanged with permutation
- other tasks such as scene recognition/segmentation, such order matters

[1] Deep Sets, Zaheer et al., NIPS2017

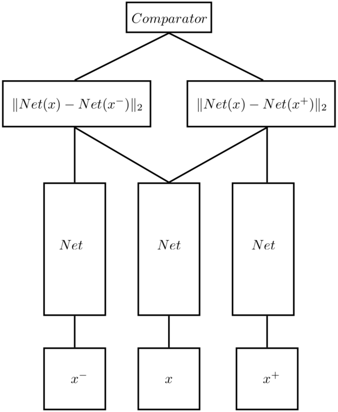


Figure 1: Triplet network structure

- change the Net with PointNet structure, then for experiment[2]

[2] ref: Deep Metric Learning Using Triplet Network, ICLR2015

Points Cloud Sampler - Future Work

- refine preprocess procedure shape retrieval with metric learning
- deep sets -
- continue contributing to the open source module, pyntcloud

[1] https://github.com/daavoo/pyntcloud

LatentGNN: with RL and 3D

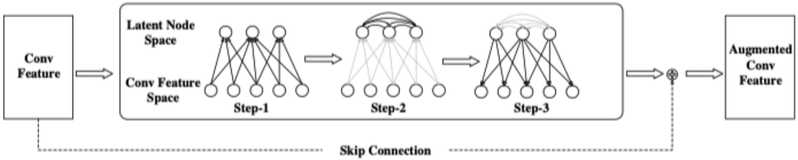
- Graph Neural Nets overview
  - fully connected $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, N nodes $v_i \in \mathcal{V}$, and edges $(v_i, v_j) \in \mathcal{E}$,
  - $\mathbf{x}_i$ is the input for node v_i - a vector of (input channels, 1)
  - update via an iterative message passing process

$$\tilde{\mathbf{x}}_i = h\left(\frac{1}{Z_i(\mathbf{X})}\sum_{j=1}^{N} g\left(\mathbf{x}_i, \mathbf{x}_j\right)\mathbf{W}^\top \mathbf{x}_j\right) \tag{1}$$

-
  - $\tilde{\mathbf{x}}_i \rightarrow$ updated feature, $h(\cdot) \rightarrow$ activation function(ReLU), $g(\cdot,\cdot) \rightarrow$ kernel function, $\mathbf{W} \rightarrow$ weight matrix, $\mathbf{Z}_i(\mathbf{x}) \rightarrow$ normalization factor

[1] LatentGNN: Learning Efficient Non-local Relations for Visual Recognition, ICML(Oral) 2019

[2] basically the input tensor for the actual training is 4D: (**b**atch, **w**eight, **h**eight, **c**hannels), the input channels can be RGB, Depth, XYZ, Normal, Local Point Density, Local Curvature or Learned Features, or any combination, just concatenate them together



- divide such iteration into three steps:
  - step 1 visible-to-latent propagation: $\mathbf{z}_k = \sum_{j=1}^{N} \psi(\mathbf{x}_j, \theta_k) \mathbf{W}^\top \mathbf{x}_j$
  - step 2 latent-to-latent propagation: $\mathbf{z}_k = \sum_{j=1}^{N} \psi(\mathbf{x}_j, \theta_k) \mathbf{W}^\top \mathbf{x}_j$
  - step 3 latent-to-visible propagation: $\tilde{\mathbf{x}}_i = h\left(\sum_{k=1}^{d} \psi(\mathbf{x}_i, \theta_k) \tilde{\mathbf{z}}_k\right)$
- after iterations, then augment the feature $\mathbf{X}$ with a weighted summation as in the ResNet:

$$\mathbf{X}_{aug} = \lambda \tilde{\mathbf{X}} + \mathbf{X} \tag{2}$$

[1] LatentGNN: Learning Efficient Non-local Relations for Visual Recognition, ICML(Oral) 2019
[2] Deep Residual Learning for Image Recognition, CVPR(Oral) 2016

LatentGNN: Our Module

- outperform the backbones, both 2D(MaskRCNN on MSCOCO) and 3D(PointNet++ on ScanNet)
- more complex, increase in the FLOPS and #Params

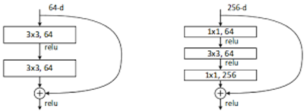
Model	Kernels	Scale	Pixel Accuracy	Voxel Accuracy	Class Pixel Accuracy	Class Voxel Accuracy	FLOPS	#Params
3DCNN(Dai et al., 2017a)	-	-	-	73.0	-	-	-	-
PointNet(Qi et al., 2017a)	-	-	-	73.9	-	-	-	-
PointCNN(Li et al., 2018)	-	-	85.1	-	-	-	-	
PointNet++(Qi et al., 2017b)	-	Single Scale	81.5	83.2	51.7	53.1	-	-
PointNet++(Qi et al., 2017b)	-	Multi Scale	-	84.5	-	-	-	-
+NL Block	1	Single Scale	82.3	84.0	53.1	54.5	+31M	+0.70M
+LatentGNN	1	Single Scale	82.6	84.2	53.2	54.6	**+15M**	**+0.31M**
+LatentGNN	3	Single Scale	**83.7**	**85.2**	**56.0**	**57.6**	+30M	+0.54M

Table 3. Performance of LatentGNN on **ScanNet** based on PointNet++.

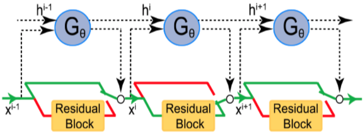
[1] LatentGNN: Learning Efficient Non-local Relations for Visual Recognition, ICML(Oral) 2019
[2] PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space,NeurIPS 2017

LatentGNN: Recurrent and Skip

- main contribution and novelty of our project
  - recurrent on the latent iterative module(my part)
    - similar as the ResNet
    - has shown improvements with empirical trials
  - skip diagram to save the calculations, aka FLOPS(yutong's part)
    - similar as the SkipNet, in the general frame of AutoML



(a) ResNet Basis



(b) SkipNet Basis

LatentGNN: future work

- merge the pipelines together
- theoretical proof and empirical results

The End