Directional Graph Networks with Hard Weight Assignments

Miguel Dominguez and Raymond Ptucha (from Rochester Institute of Technology)

Introduction

- Point cloud networks emulate traditional CNNs
- Map learnable weight matrices to points in a neighborhood
- This mapping may not be 1:1
- Other works create soft mapping between every weight matrix and every neighbor: $O(K^2)$
- A hard mapping $O(K)$ performs better

Method

- Map each weight matrix to angular (directional) regions on a sphere
- Generate KNN graph
- Select one edge per weight matrix in each neighborhood which is either in angular region or as close as possible
- Perform weighted sum on the chosen edges

Results

- Above: Visualization of 1st layer filters
- Below: Overall results on Benchmarks

<table>
<thead>
<tr>
<th>Network</th>
<th>MN40 (1k)</th>
<th>ShapeNet</th>
<th>SHREC’15</th>
</tr>
</thead>
<tbody>
<tr>
<td>PointNet++</td>
<td>91.9%</td>
<td>85.1</td>
<td>94.1%</td>
</tr>
<tr>
<td>DGCNN</td>
<td>92.9%</td>
<td>85.2</td>
<td>-</td>
</tr>
<tr>
<td>SpiderCNN</td>
<td>92.4%</td>
<td>85.3</td>
<td>95.8%</td>
</tr>
<tr>
<td>PointCNN</td>
<td>92.2%</td>
<td>86.1</td>
<td>-</td>
</tr>
<tr>
<td>HDGN (Ours)</td>
<td><strong>93.9%</strong></td>
<td><strong>85.4</strong></td>
<td><strong>100%</strong></td>
</tr>
</tbody>
</table>

Ablations

- Left: $O(K^2)$ Soft assignment filters
- Right: $O(K)$ Hard assignment filter

• Top: More directional weight matrices improves accuracy
• Bottom: Directional weight matrices are sensitive to angle if not trained on rotational variation

Conclusion

- First step in reducing complexity of point cloud networks that use graphs
- Future work will address the large overhead to store graph information