LidarNAS: Unifying and Searching Neural Architectures for 3D Point Clouds
Chenxi Liu Zhaoqi Leng Pei Sun Shuyang Cheng Charles R. Qi Yin Zhou Mingxing Tan Dragomir Anguelov
Waymo LLC

Motivation
Observation:
- Neural architectures for 3D point clouds exhibit a large variety
- Diverse set of concepts in architecture names: PointNet, VoxelNet, PointPillars, Range Sparse Net,...
- This level of variety is not observed in 2D images
Sources of Variety:

2D images | 3D point clouds
---|---
Views | perspective, unordered set, top-down,...
Sparsity | dense, sparse
Layers | conv2d, mlp, conv2d, sparse conv2d, sparse conv3d,...

Our Goal:
- A unified framework that can interpret and organize the variety of neural architecture designs
- Materialize this framework into an architecture search space, which unlocks and enables a principled Neural Architecture Search for 3D
- Demonstrate improved performance as well as interesting lessons about neural architectures for 3D

Unify Neural Architectures for 3D
Philosophy
- Despite the variety on the surface, the underlying principle is surprisingly congruent: finding some neighborhood of the 3D points and then aggregating information within.
  - “neighborhood” =
    - Euclidean ball (PointNet++)
    - 3D neighborhood from Cartesian (x, y, z) (VoxelNet)
    - 2D neighborhood from Cartesian (x, y) (PointPillars)
    - 2D neighborhood from pixel index (i, j) (LaserNet)
- “aggregation” = some form of convolution / pooling
- Different data views can transform between each other back and forth. However, once the data view is determined, it restricts the type of layers that can be applied.

Key Concepts
- Views and formats (6): Point, Pillar, Pillar (sparse), Voxel (sparse), Perspective (sparse)
- Transforms ($6^2 = 36$): From one view-format combination to another
- Layers: Depending on the view-format combination
- Stages: Each one = sequential pair of possible transforms and their associated layers. Entire backbone = $S$ stages.

LidarNAS Framework / Search Space

Inclusion of Existing Designs

(a) Multi-View Fusion (b) Sparse Point-Voxel

Search Neural Architectures for 3D

From Framework to Search Space
- Transforms
  - No pillar to voxel. From voxel, only to pillar.
  - 31 / 36: still high coverage
- Layers
  - Point: multiple layers of dense-normalization-ReLU
  - 2D dense: U-Net with residual blocks (conv2d)
  - 2D sparse: U-Net with residual blocks (sparse conv2d)
  - 3D sparse: U-Net with residual blocks (sparse conv3d)
- Stages: $S = 3$

Search Algorithm: Regularized Evolution
Why evolutionary NAS, not weight-sharing NAS?
- Evolutionary NAS arguably makes the least approximations
- Weight-sharing NAS is too GPU memory intensive for 3D tasks, which already had a small batch size (< 10) per GPU.
First randomly select a stage, then randomly apply one of the following six mutation choices to this stage:
- Add a view: if the stage does not have all four views, then randomly add a view not yet present
- Remove a view: if the stage has more than one view, then randomly remove an existing view
- Switch the view: if the stage has exactly one view, then switch the view to another
- Adjust the pillar / voxel size: multiply by either 0.8 or 1.2
- Adjust the number of channels: multiply by either 0.8 or 1.2
- Adjust the layer progression: increase or decrease the number of dense-normalization-ReLU repeats / U-Net scales
The first four focus on “transform”; the last two focus on “layer”.

Experimental Results

Improved Detection on Waymo Open Dataset

<table>
<thead>
<tr>
<th>model</th>
<th>vehicle L1 AP</th>
<th>pedestrian L1 AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>LaserNet</td>
<td>52.1</td>
<td>64.3</td>
</tr>
<tr>
<td>PointPillars</td>
<td>63.0</td>
<td>70.0</td>
</tr>
<tr>
<td>PV-RCNN</td>
<td>70.0</td>
<td>80.0</td>
</tr>
<tr>
<td>Pillar-based</td>
<td>69.8</td>
<td>78.5</td>
</tr>
<tr>
<td>PV-RCNN</td>
<td>77.7</td>
<td>90.9</td>
</tr>
<tr>
<td>RCD</td>
<td>70.9</td>
<td>87.6</td>
</tr>
<tr>
<td>MVF++</td>
<td>74.6</td>
<td>83.1</td>
</tr>
<tr>
<td>CenterPoint</td>
<td>75.6</td>
<td>85.3</td>
</tr>
<tr>
<td>PPC</td>
<td>65.2</td>
<td>80.8</td>
</tr>
<tr>
<td>RangeDet</td>
<td>72.9</td>
<td>89.9</td>
</tr>
</tbody>
</table>

Comparison of Warm Start and Evolved Architectures

LidarNASNet-P 1
LidarNASNet-R 1

Lessons from Sampled Architectures
- Search space is non-trivial and challenging (bottom left figure)
- Mutating “transforms” results in larger performance changes than mutating “layers” only
- Later stages matter more; top-down views (voxel and pillar) influence detection AP positively, while perspective view negatively (bottom right figure)
- Sparse ≠ fast: More sparse branches result in smaller latency if pillar view, but larger latency if perspective view