Traditional methods for solving point cloud classification and segmentation reply on handcrafted features to capture geometric properties of the point cloud. The task of designing deep neural network suitable for 3D data is non-trivial. For example, most information in RGB images is encoded as pixels and pixels intensity distribution for each color channel. However, information of 3D models resides on the surface, unlike 2D pixels images. Therefore, feature useful for 2D image classification might not necessarily be sufficient for 3D model’s classification. CAD models are often colorless and therefore the models must be able to pick up on other features. Standard deep neural network models take as input data with regular structures while point clouds are fundamentally irregular; point positions are continuously distributed in the space and any permutation of their ordering does not change the spatial distribution.

One common approach to process point cloud data using deep learning models is to first convert raw point cloud data into volumetric representation, namely 3D grid. This approach, however, usually introduces quantization artifacts and excessive memory usage making it difficult to go to capture high-resolution or fine-grained.

**Point Net:**

Typical convolutional architecture requires highly regular input data formats like image gird or 3d voxels in order to perform weight sharing and other kernel optimization. Since point cloud are not in a regular format, an easy way is to transform such data to regular 3d voxel grid or collection of images (views) before feeding them to a deep net architecture. This date representation transformation renders the resulting data unnecessarily voluminous while also introduces quantization artifacts that can obscure natural invariances to the data.

For this reason, we focus on a different input representation for 3D geometry using simply point clouds. Point cloud are simple and unified structures that avoid the combinatorial irregularities and complexities of meshes and thus easier to learn from. However, when dealing with point cloud, one should respect the fact that a point cloud is set of points and therefore invariant to permutations of it members, necessitating certain symmetrizations in the net computation. Further invariance to rigid motions also need to be considered.