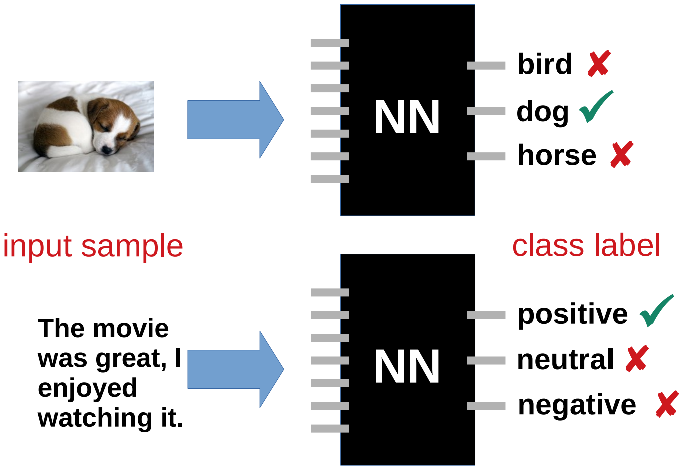
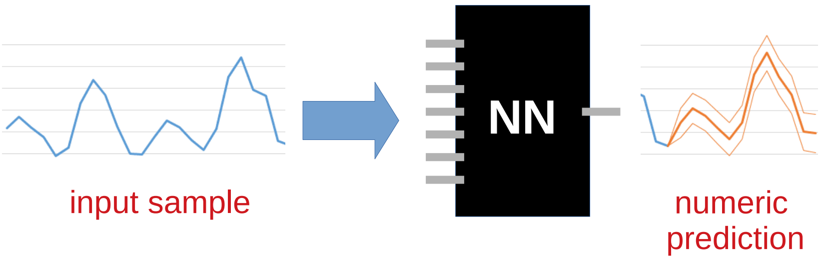
Symmetry detection using neural networks

Damjan Strnad

Neural networks 101

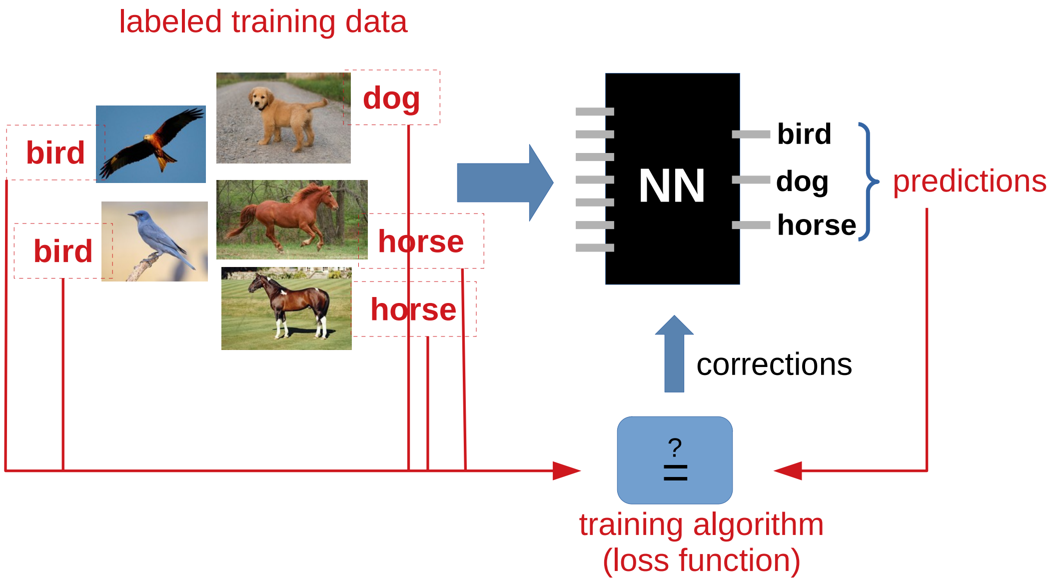
- think of it as a programmable chip
- for a given numerical input (image, text embedding, point cloud) produces numerical output (class probability, numeric value prediction)





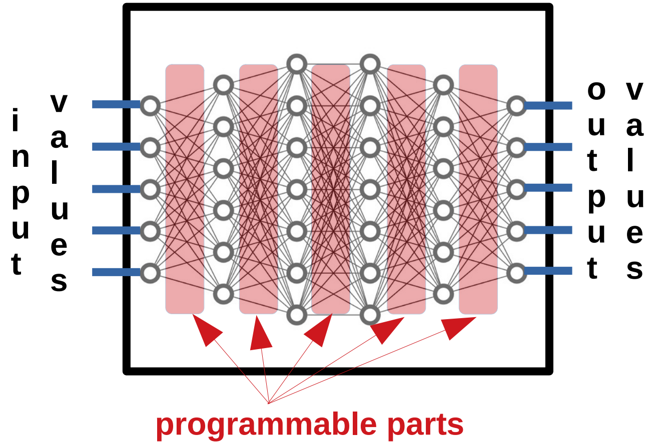
Neural networks 101

- programming the chip = training the neural network
- training is supervised ⇒ training dataset contains training samples with given ground truth label/value



Neural networks 101

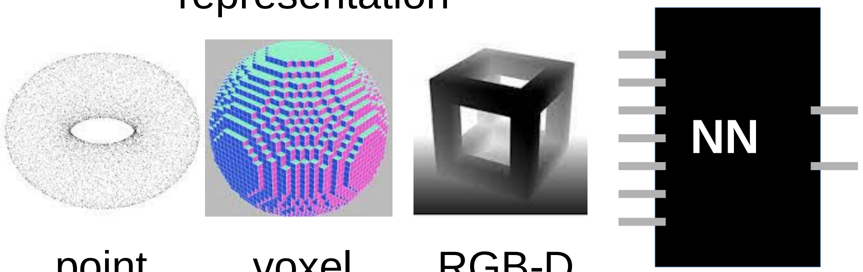
- NN performs a non-linear mapping of input **x** into output **f**(**x**) through multiple transformation layers
- different types of layers for different purposes (fully connected, (de)convolutional, (un)pooling,...)
- training adjusts the weights on internal connections between layers



Neural networks for symmetry detection

- goal: determine reflection plane and/or rotation axis + angle for a given 3D input object

3D object representation



plane coefficients
rotation axis + angle
} repeated for a fixed number of planes/axes

point cloud | voxel grid | RGB-D image

- training can be supervised (we provide target values directly) or unsupervised (we describe the loss in terms of some matching error when performing reflection/rotation)

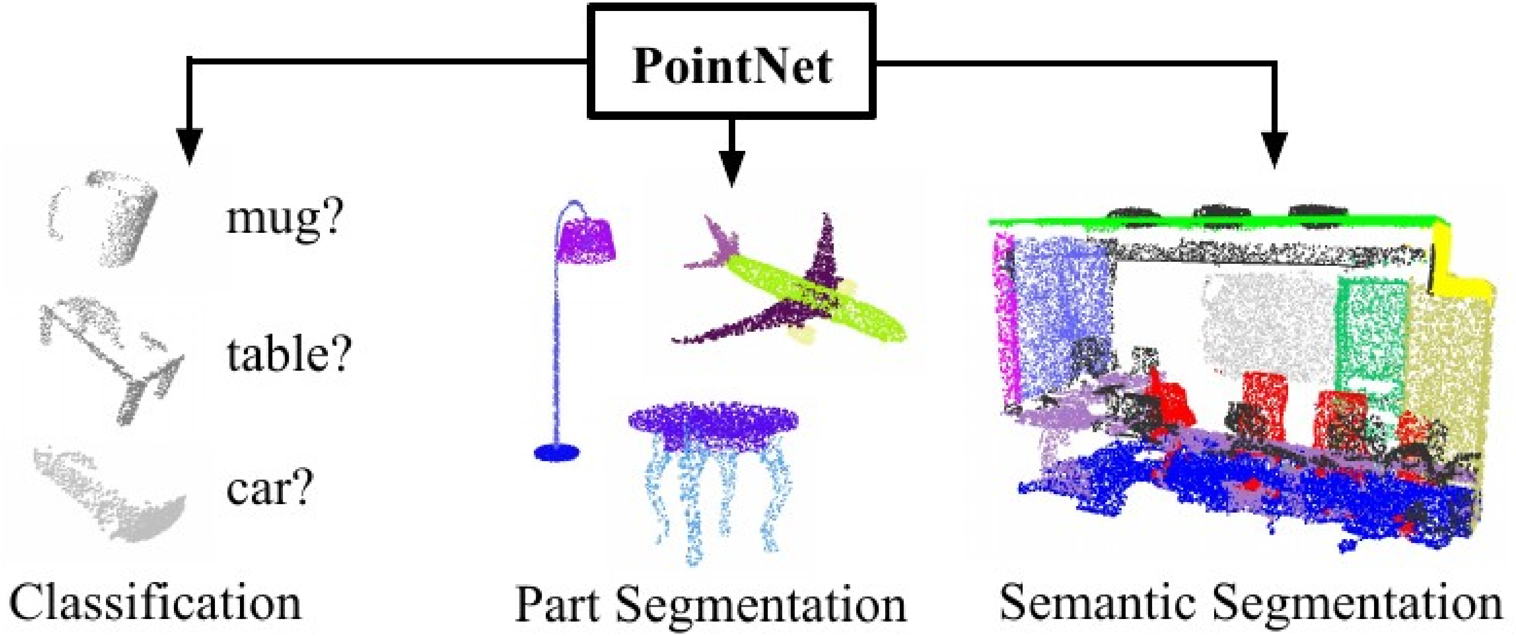
Neural networks for 3D data

- typical NN architectures require structured input (images, voxel grids, sequential/ordered data)
- point clouds are unstructured and unordered, just a list of points
- we want the same result regardless of point order
  - when predicting a single value for the whole cloud, we need **permutation invariance**
  - when predicting a value per input point, we need **permutation equivariance**

Equivariance in NN

- a layer is equivariant, if its outputs change in correspondence with the permutation of inputs (e.g., segmentation of points in the input cloud)
- a layer is invariant if its output is constant with respect to the permutation of inputs (e.g. classification of input object)
- reference neural network architectures – PointNet and PointNet++

PointNet[1] – applications

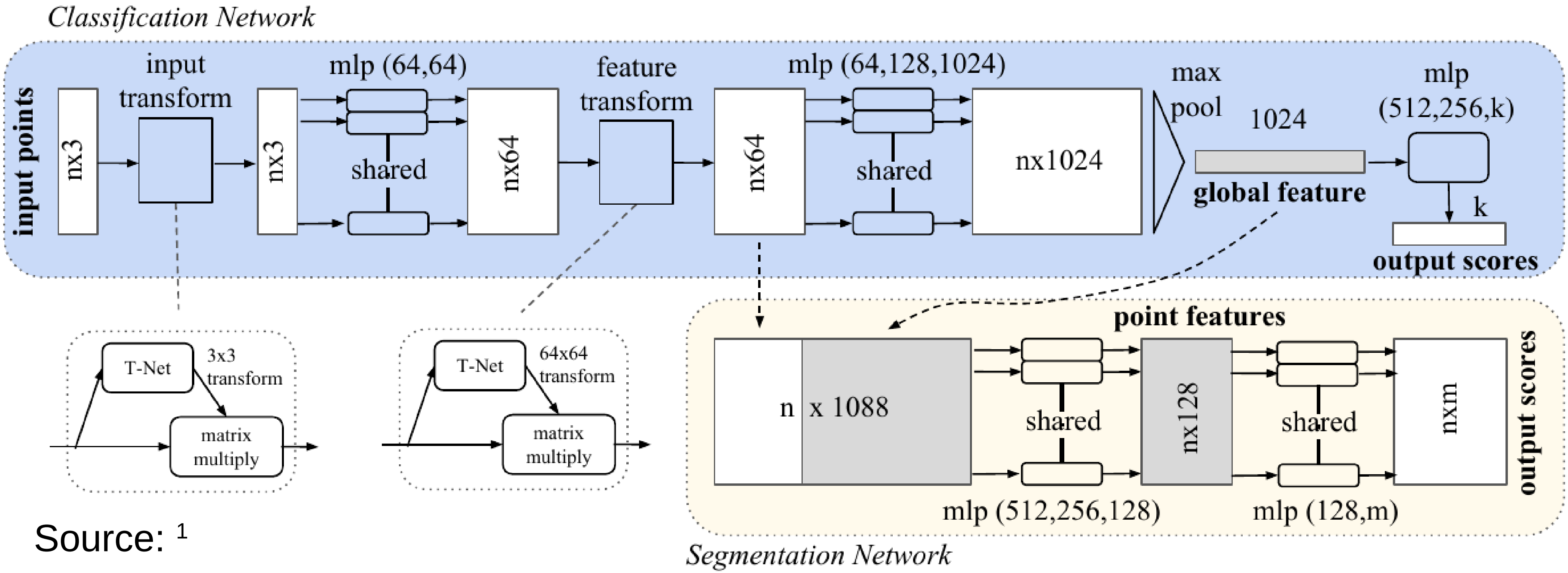


Source: [1]

[1]Qi C., et al., *PointNet - Deep Learning on Point Sets for 3D Classification and Segmentation*, CVPR 2017, arXiv:1612.00593.

PointNet[1] – architecture

- classification and segmentation part share a lot of structure[1]

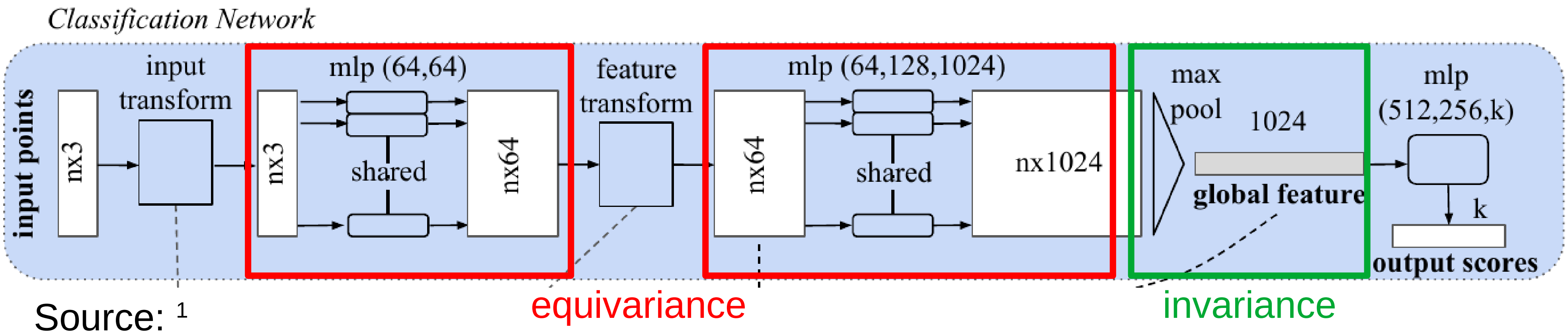


Source: [1]

[1]Qi C., et al., *PointNet - Deep Learning on Point Sets for 3D Classification and Segmentation*, CVPR 2017, arXiv:1612.00593.

PointNet[1] – architecture

- each input point is independently processed by a shared MLP (1D convolution), which extracts *M* features per point
- the MLP weights are shared ⇒ equivariance is achieved
- max pooling extracts the maximum value over all points per feature ⇒ invariance is achieved



Source: [1]

[1]Qi C., et al., *PointNet - Deep Learning on Point Sets for 3D Classification and Segmentation*, CVPR 2017, arXiv:1612.00593.

PointNet[1] – principle

- the network learns to extract informative points from the cloud – functions *h* and MAX
- the attributes of selected points are used to perform object classification or segmentation (per point classification) – MLP function *γ*

$$f(x_1, x_2, ..., x_n) = \gamma\left(\underset{i=1,\dots,n}{\mathrm{MAX}}\{h(x_i)\}\right)$$

- the input cloud is transformed using a learned rigid transformation (T-net – network within a network)

[1]Qi C., et al., *PointNet - Deep Learning on Point Sets for 3D Classification and Segmentation*, CVPR 2017, arXiv:1612.00593.

PointNet++[2]

- problem with PointNet: it doesn't capture the local cloud structure at different scales
- the idea of PointNet++:
  1) hierarchical division of input point cloud into overlapping regions (farthest point sampling)
  2) local feature extraction from regions using PointNet
  3) aggregation of local features into global ones through multiple network layers

[2]Qi C., et al., *PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space*, NIPS 2017, arXiv:1706.02413.

Neural networks for symmetry detection

- Ji P., Liu X., *A fast and efficient 3D reflection symmetry detector based on neural networks*, Multimedia Tools and Applications, 2019.
- Gao L., et al., *PRS-Net: Planar Reflective Symmetry Detection Net for 3D Models*, IEEE Trans. on Vis. and Comp. Graphics, 2021.
- Shi Y., et al., *SymmetryNet: Learning to Predict Reflectional and Rotational Symmetries of 3D Shapes from Single-View RGB-D Images*, ACM Trans. on Graphics, 2020.

Reflectional symmetry in point clouds[3]

Concept:

1) PointNet++ is used to determine which points in the input cloud belong to the symmetry plane

2) RANSAC and least squares method is used to set the initial plane equation using the points selected in step 1

3) iterative closest point (ICP) method is used to fine-tune the plane equation

[3]Ji P., Liu X., A fast and efficient 3D reflection symmetry detector based on neural networks, Multimedia Tools and Applications, 2019.

Network training

- training data are point clouds with known symmetry plane
  - points less than δ from the plane (1% of bounding box diagonal length) are labeled as 1, the rest as 0
- the number of input points is 2048, 4096, 8192 or 12288, the number of channels is 6 (coordinates + normal)
- PointNet++ is used to extract features at different scales
- the imbalance of 0 (Y^-) and 1 (Y^+) samples is solved by the weighted cross-entropy loss function:

$$L = -\alpha \sum_{j \in Y^+} \log\left(\boxed{P(y_j = 1)}\right) - (1-\alpha) \sum_{j \in Y^-} \log\left(\boxed{P(y_j = 0)}\right) \qquad \alpha = \frac{|Y^-|}{|Y^-| + |Y^+|}$$

network outputs

[3]Ji P., Liu X., A fast and efficient 3D reflection symmetry detector based on neural networks, Multimedia Tools and Applications, 2019.

Network architecture

- input is point cloud with 6 channels per point (position+normal)
- the following sequence of layers is repeated 4 times:
  - sampling of a smaller number of points (2048, 512, 128, 32)
  - finding *K*=32 nearest neighbors per point (grouping layer)
  - three 1x1 convolution layers with increasing number of filters
  - invariant global max pooling across the neighborhood
- the following sequence of layers is repeated 4 times:
  - interpolate features from a smaller to a larger number of points
  - concatenate with features on max pooling output above with the same number of points
  - three 1x1 convolution layers with 256 filters
  - invariant global max pooling across the neighborhood
- two fully-connected layers for classification of points into 0/1 classes

[3]Ji P., Liu X., A fast and efficient 3D reflection symmetry detector based on neural networks, Multimedia Tools and Applications, 2019.

Extracting symmetry plane

- because of false positives, RANSAC is used to produce candidate planes
- for a selected point subset the plane equation is determined using least squares; it is assumed that the plane goes through the origin
- after the plane is determined, the original point cloud P is reflected over it to get point cloud Q
- ICP optimization aligns point clouds P and Q
- the rest of points that were classified as positive is used to determine additional planes of symmetry

[3] Ji P., Liu X., A fast and efficient 3D reflection symmetry detector based on neural networks, Multimedia Tools and Applications, 2019.

Results

- training with 1100 point clouds of 8192 points, randomly rotated and with added noise, takes 6 hours (GTX 980)
- classification in 70 ms, ICP in 236 ms on average
- efficiency is measured with average classification accuracy (best case 81.89%) and angle between the predicted and true plane of symmetry (best case 3.17°)
- higher number of input points and normal information improve classification, ICP improves plane prediction
- limitation: global symmetry only, symmetric object assumed

[3]Ji P., Liu X., A fast and efficient 3D reflection symmetry detector based on neural networks, Multimedia Tools and Applications, 2019.

PRS-Net[4]: Reflection and rotation symmetry in voxel grids

Concept:

1) network input is voxelized geometry

2) a sequence of convolutions and max pooling layers produces a feature vector of size 64

3) three separate branches of fully connected layers predict from the feature vector at most three planes of reflection symmetry (plane parameters ($\mathbf{n}$,d)) and three axes of rotation symmetry (quaternion $\mathbf{p}_i=(u_{0i},(u_{1i},u_{2i},u_{3i}))$)

4) the unsupervised loss function is a symmetry metric that reflects/rotates geometry using predicted plane/axis

[4]Gao L., et al., PRS-Net: Planar Reflective Symmetry Detection Net for 3D Models, IEEE TVCG, 2021, arXiv:1910.06511.

PRS-Net[4] – loss function

$$\mathbf{q}'_k = \mathbf{q}_k - 2\frac{\mathbf{q}_k \cdot \mathbf{n}_i + d_i}{\|\mathbf{n}_i\|^2}\mathbf{n}_i$$

reflection map

$$\hat{\mathbf{q}}'_k = \mathbf{p}_j \hat{\mathbf{q}}_k \mathbf{p}_j^{-1}$$

rotation map

Two parts:

1) symmetry distance loss:

- input object O is uniformly sampled and N points are mapped using reflection/rotation
- calculate smallest distance D_k of mapped k=1..N points from O (for speed-up the distance to centers of uniform grid are used)
- the loss value L_{sd} is a sum over all points, planes and axes

$$D_k = \min_{\mathbf{o} \in O} \|\mathbf{q}'_k - \mathbf{o}\|_2$$

$$L_{sd} = \sum_{i=1}^{3}\sum_{k=1}^{N} \hat{D}_k^{(i)} + \sum_{j=1}^{3}\sum_{k=1}^{N} \tilde{D}_k^{(j)}$$

PRS-Net[4] – loss function

Two parts:

2) regularization:

- prevents predicting the same plane/axis on all outputs
- unit normals of planes/axes are columns of matrices M_1 and M_2
- compute A and B, which are 0 in case of orthogonal planes/axes
- define regularization penalty L_r and total loss L

$$\mathbf{A} = \mathbf{M}_1\mathbf{M}_1^T - \mathbf{I}, \qquad L_r = \|\mathbf{A}\|_F^2 + \|\mathbf{B}\|_F^2 = \sum_{i=1}^{3}\sum_{j=1}^{3}(\mathbf{A}_{ij}^2 + \mathbf{B}_{ij}^2),$$

$$\mathbf{B} = \mathbf{M}_2\mathbf{M}_2^T - \mathbf{I}, \qquad L = L_{sd} + w_r L_r,$$

[4]Gao L., et al., PRS-Net: Planar Reflective Symmetry Detection Net for 3D Models, IEEE TVCG, 2021, arXiv:1910.06511.

PRS-Net[4] – validation

- the network predicts three planes and three axes
- if actual planes are <3, the predictions will repeat
  - for plane/axis pairs with dihedral angle $<\pi/6$, the one with larger error is removed
- remove also predictions, which are poor local minima
  - planes with $L_{sd} > 4\text{x}10^{-4}$ are removed
- only global rotation symmetry is detected ⇒ remove all axes for which rotation by 1° is not symmetric

[4] Gao L., et al., PRS-Net: Planar Reflective Symmetry Detection Net for 3D Models, IEEE TVCG, 2021, arXiv:1910.06511.

PRS-Net[4] – results

- ShapeNet dataset, 51300 objects, using random rotations the training set is augmented to 4000 objects per each of 55 classes; manual selection of symmetric objects and ground truth determination
- sample 1000 points/object and generate 32^3 voxel grid
- PRS-Net is compared to 7 other methods
- for comparison, GTE and SDE (identical to L_{sd}) between the predicted plane and GT are used:

$$GTE = (a_i - a_{gt})^2 + (b_i - b_{gt})^2 + (c_i - c_{gt})^2 + (d_i - d_{gt})^2.$$

[4] Gao L., et al., PRS-Net: Planar Reflective Symmetry Detection Net for 3D Models, IEEE TVCG, 2021, arXiv:1910.06511.

PRS-Net[4] – results

- comparison on asymmetric objects from datasets SCAPE, ABC and Thingi10k (GTE ≈ 0.001, SDE ≈ 0.000086)
- they demonstrate application to shape completion
- robustness is tested by adding noise and removing object parts
- testing is performed using different voxel grid resolutions (optimal is 32^3)
- supervised network pretraining, where a related method is used to provide target values, halves training time

[4]Gao L., et al., PRS-Net: Planar Reflective Symmetry Detection Net for 3D Models, IEEE TVCG, 2021, arXiv:1910.06511.

SymmetryNet[5]: Reflection and rotation symmetry from RGB-D images

Basis:

- addresses symmetry detection with missing data (e.g., owing to occlusion in RGB-D images from single viewpoint)
- in such cases, instead of purely geometric symmetry detection, it is useful to apply data-driven statistical prediction
- humans determine symmetry of known objects based on experience; in unknown objects we look for signs of local symmetry on the visible part of an object and imagined invisible part of the object – shape matching and completion all at once

[5]Shi Y., et al., SymmetryNet: Learning to Predict Reflectional and Rotational Symmetries of 3D Shapes from Single-View RGB-D Images, ACM Trans. on Graphics, 2020, arXiv:2008.00485.

SymmetryNet[5]: Reflection and rotation symmetry from RGB-D images

Basis:

- direct prediction of symmetry is prone to overfitting (the model memorizes object symmetries and performs object recognition) – the solution is multi-task learning:
  - the network predicts reflection symmetry planes and cylindrical rotation symmetry axes (continuous (revolutions) and discrete)
  - at the same time the network matches symmetric counterparts to all points (probability heatmap for each other point to be the symmetric counterpart, and its *xyz* coorinates)

[5] Shi Y., et al., SymmetryNet: Learning to Predict Reflectional and Rotational Symmetries of 3D Shapes from Single-View RGB-D Images, ACM Trans. on Graphics, 2020, arXiv:2008.00485.

SymmetryNet[5]: Reflection and rotation symmetry from RGB-D images

Concept:

- the network predicts multiple symmetries of particular type:
  - M_{refl} reflection symmetries and M_{rot} rotation symmetries as pairs $(\mathbf{p}_i, \mathbf{n}_i)$, where **p** is the point on symmetry plane/axis and **n** its normal/direction
  - all symmetries are expressed in camera coordinate system
- predictions at network output are paired with corresponding ground truths using optimal assignment

[5]Shi Y., et al., SymmetryNet: Learning to Predict Reflectional and Rotational Symmetries of 3D Shapes from Single-View RGB-D Images, ACM Trans. on Graphics, 2020, arXiv:2008.00485.

SymmetryNet[5] – architecture

- three main components:

1. extraction of point-wise features from RGB-D
    - appearance features are extracted from RGB using CNN, geometry features from depth image using PointNet
    - besides point features, global features are considered: instead of average or max pooling they use spatially weighted pooling
2. use of point+global features for point-wise symmetry prediction
    - 3-layer MLP is used for prediction (exact architecture is not given)
    - each point predicts multiple symmetry planes/axes
3. point-wise predictions are aggregated and filtered

[5]Shi Y., et al., SymmetryNet: Learning to Predict Reflectional and Rotational Symmetries of 3D Shapes from Single-View RGB-D Images, ACM Trans. on Graphics, 2020, arXiv:2008.00485.

SymmetryNet[5] – principle

- outputs for multi-task learning for each point:
  - symmetry type classification (0 = null, 1 = reflection, 2 = rotation)
  - regression of symmetry parameters **p** and **n**
  - regression of symmetric counterpart location
  - for rotation symmetry, the symmetry order is additionally predicted (r=0 for continuous, r>0 for discrete symmetry); maximum order is R=10 (number of network outputs)

[5] Shi Y., et al., SymmetryNet: Learning to Predict Reflectional and Rotational Symmetries of 3D Shapes from Single-View RGB-D Images, ACM Trans. on Graphics, 2020, arXiv:2008.00485.

SymmetryNet[5] – loss function

- loss function: $\mathcal{L} = \frac{1}{N}\sum_i^N \mathcal{L}_i$, where per point loss: $\mathcal{L}_i = \mathcal{L}_i^{\text{type}} + \mathcal{L}_i^{\text{sym}}$
- L_i^{type} is cross-entropy error for symmetry type classification
- L_i^{sym} is regression loss for symmetry parameters:

$$\mathcal{L}_i^{\text{sym}} = \begin{cases} \mathcal{L}_i^{\text{ref\_reg}} + w^{\text{ref}} \cdot \mathcal{L}_i^{\text{ref\_cp}}, & \text{if ref. sym.} \\ \mathcal{L}_i^{\text{rot\_reg}} + w^{\text{rot}} \cdot \mathcal{L}_i^{\text{rot\_cp}}, & \text{if rot. sym.} \\ 0, & \text{if no sym.} \end{cases}$$

  - $\mathcal{L}_i^{\text{ref\_reg}}$ is regression loss for predicted symmetry plane/axis
  - $\mathcal{L}_i^{\text{ref\_cp}}$ is symmetric counterpart loss
  - w^{ref} and w^{rot} are weights

[5]Shi Y., et al., SymmetryNet: Learning to Predict Reflectional and Rotational Symmetries of 3D Shapes from Single-View RGB-D Images, ACM Trans. on Graphics, 2020, arXiv:2008.00485.

SymmetryNet[5] – prediction

- predicting arbitrary number of symmetries in arbitrary order:
  - every input point produces M^{ref} candidates for reflection and M^{rot} candidates for rotation symmetry; the classifier determines the presence
  - for present symmetries, the Hungarian algorithm finds the best match:

$$\arg\max_{\Pi} \sum_{m=1}^{M} \sum_{k=1}^{K} B_{m,k} \Pi_{m,k} \quad \text{s.t.} \sum_{m=1}^{M} \Pi_{m,k} = 1, k \in \{1 \ldots K\}; \sum_{k=1}^{K} \Pi_{m,k} \leq 1, m \in \{1 \ldots M\}$$

    - Π is permutation matrix, where $\Pi_{m,k} \in \{0,1\}$ indicates, whether k-th GT symmetry is matched with m-th predicted symmetry
    - B is benefit matrix, where $B_{m,k}$ determines the »benefit« of matching k-th GT symmetry with m-th predicted symmetry

[5]Shi Y., et al., SymmetryNet: Learning to Predict Reflectional and Rotational Symmetries of 3D Shapes from Single-View RGB-D Images, ACM Trans. on Graphics, 2020, arXiv:2008.00485.

SymmetryNet[5] – inference

- point-wise symmetry predictions are aggregated:
  - DBSCAN clustering of predictions is performed and cluster centroids are returned as final predictions
  - individual point-wise prediction are weighted by $\mathcal{L}_i^{\text{type}}$
- final prediction verification:
  - RGB-D is voxelized to determine occupied, free, and invisible voxels from the viewpoint
  - the surface is transformed using the predicted symmetry
  - the error is the overlap between transformed surface points and known free regions
  - symmetries with large error are removed

[5] Shi Y., et al., SymmetryNet: Learning to Predict Reflectional and Rotational Symmetries of 3D Shapes from Single-View RGB-D Images, ACM Trans. on Graphics, 2020, arXiv:2008.00485.

SymmetryNet[5] – results

- RGB-D training set is generated from ShapeNet, YCB in ScanNet datasets
- ground truth is generated using existing methods
- training times on the order 1-3 days (Nvidia TITAN V)
- inference, aggregation and verification times on the order of 10 ms
- results are compared against three baseline symmetry detection methods for RGB-D

[5]Shi Y., et al., SymmetryNet: Learning to Predict Reflectional and Rotational Symmetries of 3D Shapes from Single-View RGB-D Images, ACM Trans. on Graphics, 2020, arXiv:2008.00485.

SymmetryNet[5] – results

- evaluation metric is PR curve, where TP and FP are determined using the error between predicted and actual symmetry:

$$\mathcal{E}^{\mathrm{ref}} = \frac{1}{N}\sum_{i}^{N}\frac{\left\|T^{\mathrm{ref}}(P_i) - \hat{T}^{\mathrm{ref}}(P_i)\right\|_2}{\rho} \qquad \mathcal{E}^{\mathrm{rot}} = \frac{1}{|\Gamma|}\frac{1}{N}\sum_{\gamma\in\Gamma}\sum_{i}^{N}\frac{\left\|T^{\mathrm{rot},\gamma}(P_i) - \hat{T}^{\mathrm{rot},\gamma}(P_i)\right\|_2}{\rho},$$

  - P_i are object points, ρ is max. distance of P_i from symmetry plane/axis
  - error tolerance for both symmetry types is 0.25
- analysis is done with respect to occlusion level (light, medium, heavy)
- typical failures: unable to detect spherical symmetry and reflectional symmetry in hidden depth direction (e.g. cube with one visible face)

[5]Shi Y., et al., SymmetryNet: Learning to Predict Reflectional and Rotational Symmetries of 3D Shapes from Single-View RGB-D Images, ACM Trans. on Graphics, 2020, arXiv:2008.00485.