

# Lab3 MLP reconstruction

王想 2100013146

## 1 Implementation

### 1.1 Model Structure

我用 MLP 拟合一个从坐标到 SDF 值的函数，进而得到物体的隐式表达。Lab 中我采用的 MLP 模型整体结构有 6 层，中间的 4 层隐藏线性层的维度均设置为 512 维，除了最后一层之外，其他 5 层后面都接入一个激活函数 Softplus。Note: 一个 MLP 只对应一个输入的点云的隐式表达。经过实验，我发现只需要 4 层隐藏层就足以获得足够好的效果，所以我并未采用更深的网络。MLP 网络拟合的函数可以用公式表达为：

$$f : \mathbb{R}^3 \rightarrow \mathbb{R}$$

$$f(pos, \theta) = sdf(pos)$$

$\theta$  为 MLP 网络的模型参数， $pos$  为点的坐标， $sdf(pos)$  表示位置  $pos$  上的 SDF 值。

### 1.2 Loss Function

损失函数我采用论文 Implicit Geometric Regularization for Learning Shapes(IGR) 中的损失函数，定义如下：

$$\ell(\theta) = \ell_{\mathcal{X}}(\theta) + \lambda \mathbb{E}_{\mathbf{x}} (\|\nabla_{\mathbf{x}} f(\mathbf{x}; \theta)\| - 1)^2$$

$$\ell_{\mathcal{X}}(\theta) = \frac{1}{|I|} \sum_{i \in I} (|f(\mathbf{x}_i; \theta)| + \tau \|\nabla_{\mathbf{x}} f(\mathbf{x}_i; \theta) - \mathbf{n}_i\|)$$

以上两式中，函数  $f(\cdot, \theta)$  为 MLP 模型； $\mathcal{X}$  为输入的点云， $I$  为输入的点云中的点的索引构成的集合； $\mathbf{x}$  为点的坐标， $\mathbf{n}_i$  为点的法向； $\mathbb{E}(\cdot)$  表示数学期望； $\lambda, \tau$  均为超参数，这两个参数我遵循原论文的设置， $\lambda = 0.1, \tau = 1.0$ 。

但是在具体实现时原论文还加上了一项 *LatentLoss* 用于将 MLP 网络泛化到所有点云(原论文作者希望训练好一个 MLP 就能运用到很多不同的点云上)，而在我的实现中我去掉了这一项，这是因为本次 Lab 的目标是针对这一个特定的点云进行重建，在这个场景下拟合的越接近越好，所以并不需要加上促使模型泛化的 *LatentLoss*。最后得出的效果证明我这样的舍弃是没问题的。

### 1.3 Decoder

在 MLP 模型训练好后，得到的就是这个物体的隐式表达。从隐式表达解码出显示表达流程如下，对给定的分辨率  $r$ ，先生成立方体网格，即  $r^3$  个对应的网格采样点，然后将这些

点的坐标输入 MLP 网络得到对应位置上的 SDF 值，最后利用 MarchingCube 算法得到显示表达，即 Mesh 表达。

## 2 Results

在默认参数设置下，用 RTX 3050 Ti 训练需要 4min20s 左右，解码需要 10s 左右。以下是在默认参数设置下得到的结果，可以看到重建出来的网格质量较高。

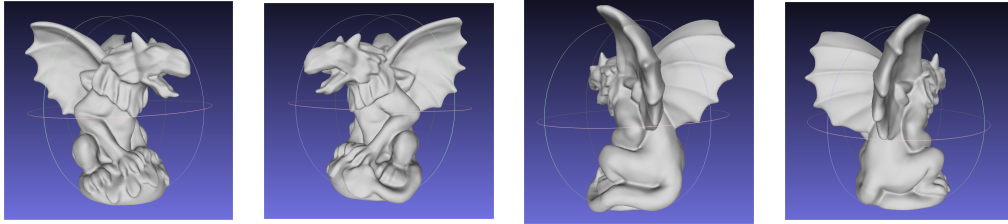


图 1: Output Mesh

## A Appendix

Requirements: *torch*, *numpy*, *scipy*, *skimage*

Running command: *python main.py --args*

-- args are given as bellow:

Perparameters	Type	Default	Help
<i>epochs</i>	int	2400	Number of trainning epochs
<i>resolution</i>	int	256	Resolution of output file
<i>load</i>	bool	True	True: load a model(need cuda); False: train a model
<i>input_path</i>	str	<i>gargoyle.xyz</i>	Path of input file
<i>output_path</i>	str	<i>output.obj</i>	Path of output file
<i>model_path</i>	str	<i>model.pt</i>	Path for load or save model

Tab 1: ArgumentParser of main.py