



深度学习及其在点云中的应用

■ 指导老师：刘斌曷

■ 报告人：李 涛



CONTENTS

01

深度学习

02

深度学习优化算法

03

卷积神经网络

04

点云数据处理

05

深度学习在点云中的应用

1.1 线性回归 (Linear Regression)

Q1: 已知某地段房屋面积、楼层、房龄等信息, 如何根据以上信息估计房屋售价?

➤ 假设:

- 自变量和因变量之间的关系是线性的;
- 噪声遵循一定的概率分布, 如正态分布。

基于线性假设, 可以将目标 (房屋售价) 表示为特征 (面积、房龄等) 的加权和, 如下式:

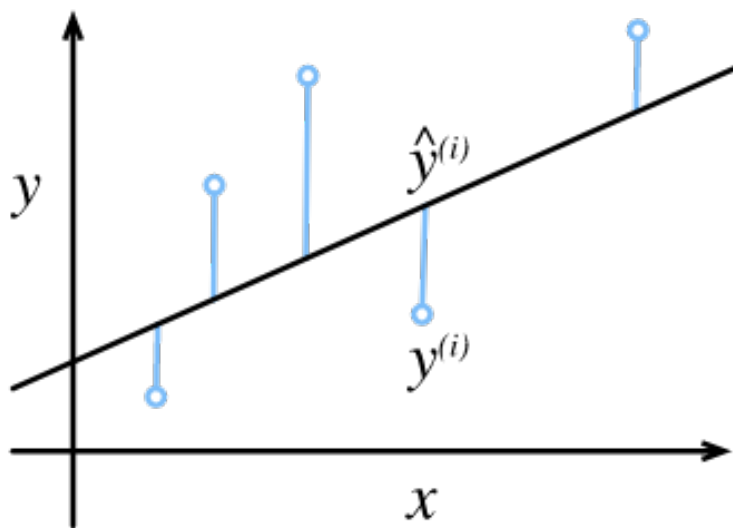
$$\hat{y} = w_1x_1 + w_2x_2 + \cdots + w_dx_d + b \quad (1-1)$$

$$\hat{y} = \mathbf{w}^T \mathbf{x} + b \quad (1-2)$$

当样本 i 的预测值为 $\hat{y}^{(i)}$, 其相应的真实标签为 $y^{(i)}$ 时, 平方误差可以定义为以下公式:

$$l^i(\mathbf{w}, b) = \frac{1}{2} (\hat{y}^{(i)} - y^{(i)})^2 \quad (1-3)$$

$$L(\mathbf{w}, b) = \frac{1}{n} \sum_{i=1}^n \frac{1}{2} (\mathbf{w}^T \mathbf{x}^{(i)} + b - y^{(i)})^2 \quad (1-4)$$



1.1 线性回归 (Linear Regression)

Q2: 如何求解模型参数 (w, b) ?

A1: 解析解

线性回归是一个简单的优化问题，只需将损失关于 w 的导数设为0，即可求得解析解：

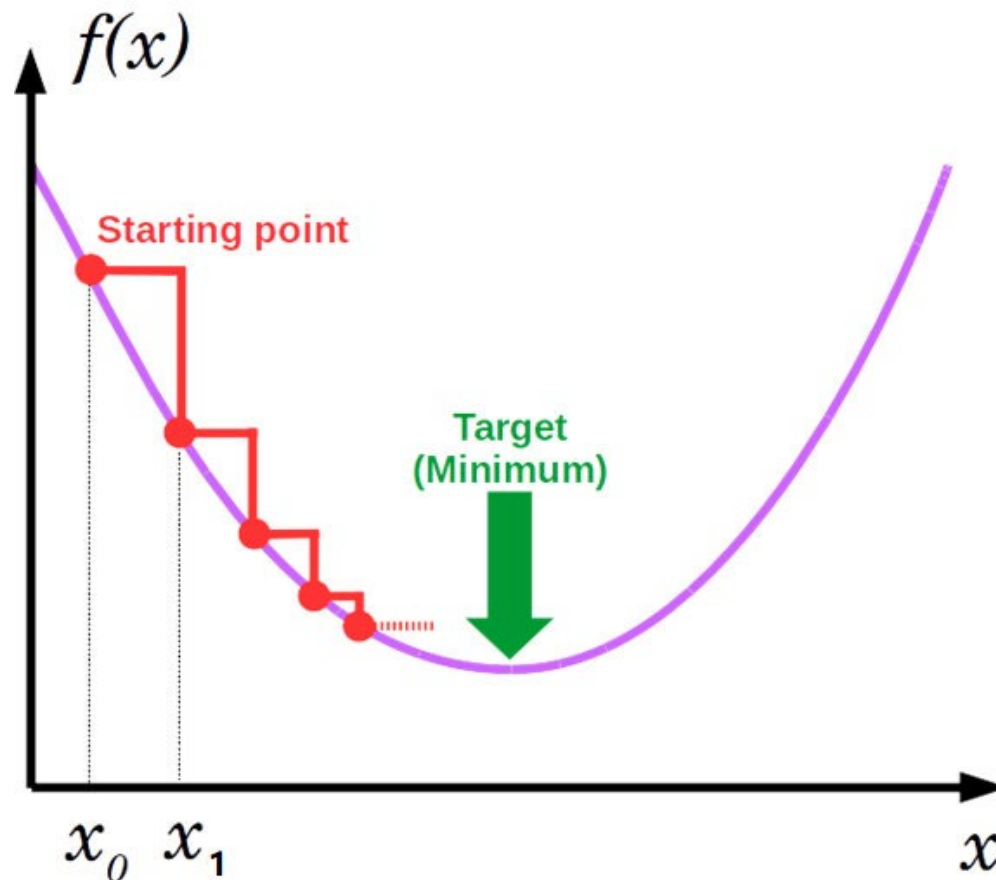
$$\mathbf{w}^* = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \quad (1-5)$$

A2: 梯度下降法

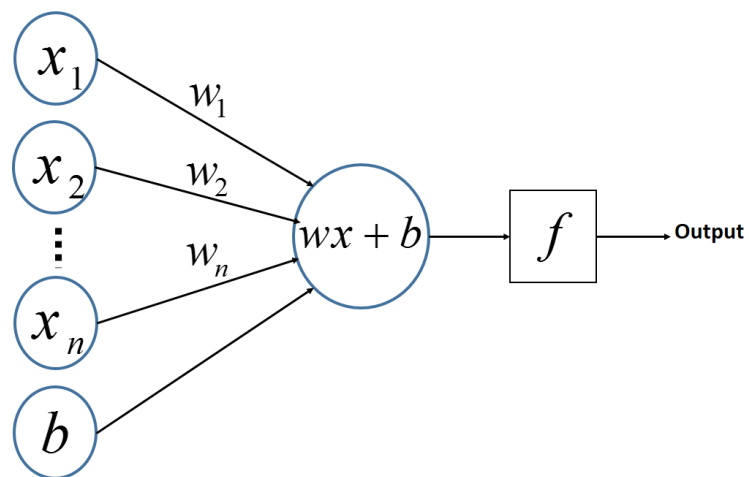
- 初始化模型参数的值，如随机初始化；
- 从数据集中随机抽取小批量样本且在负梯度的方向上更新参数，并不断迭代。

$$\mathbf{w} \leftarrow \mathbf{w} - \frac{\eta}{|\mathbf{B}|} \sum_{i \in \mathbf{B}} \partial_{\mathbf{w}} l^{(i)}(\mathbf{w}, b) \quad (1-6)$$

$$b \leftarrow b - \frac{\eta}{|\mathbf{B}|} \sum_{i \in \mathbf{B}} \partial_b l^{(i)}(\mathbf{w}, b) \quad (1-7)$$



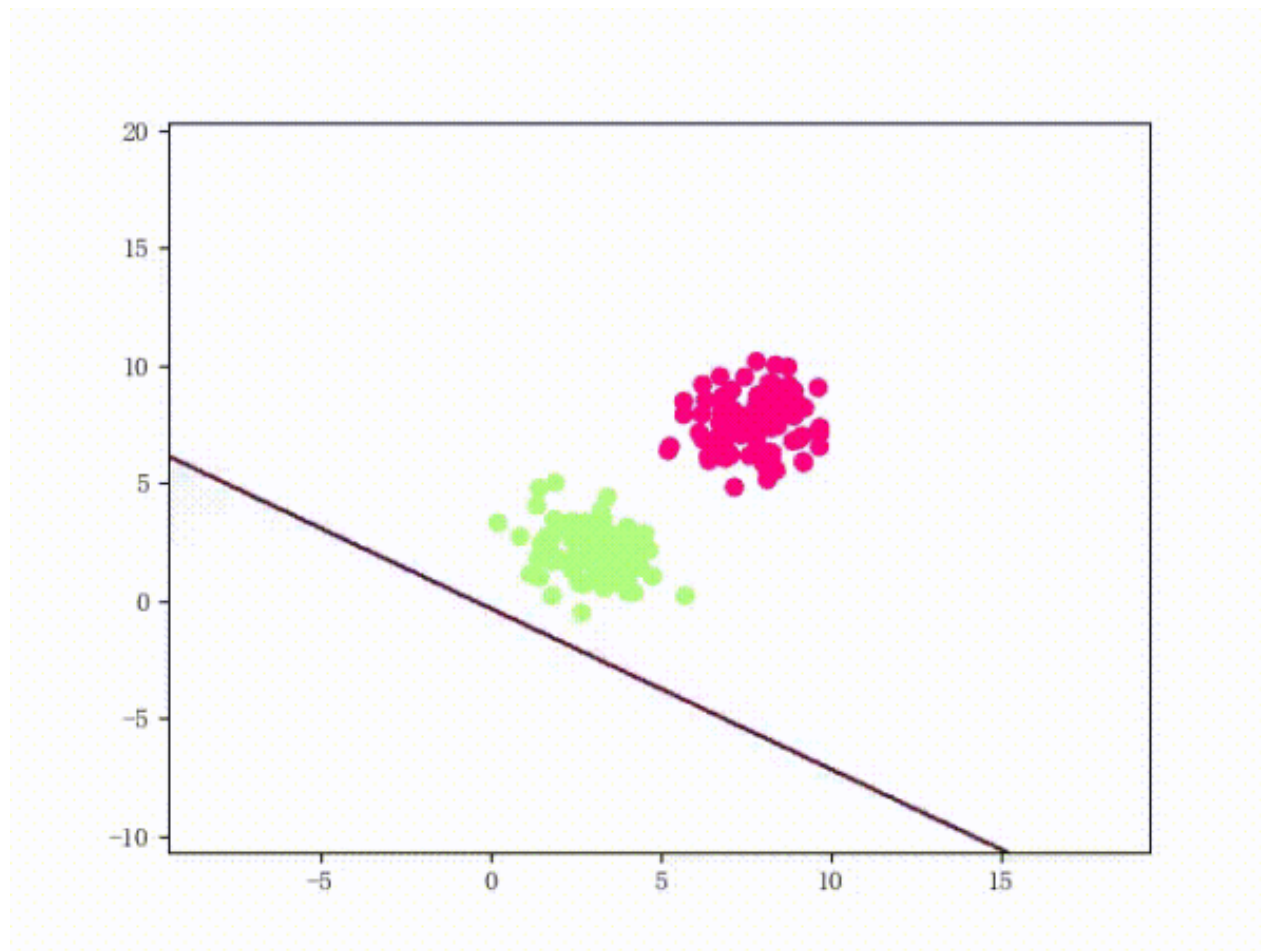
1.2 感知机 (Perceptron)



感知机的输出如下：

$$y = f(\mathbf{w}^T \mathbf{x} + b) \quad (1-8)$$

$$f(z) = \begin{cases} 0, & z < 0 \\ 1, & z \geq 0 \end{cases} \quad (1-9)$$

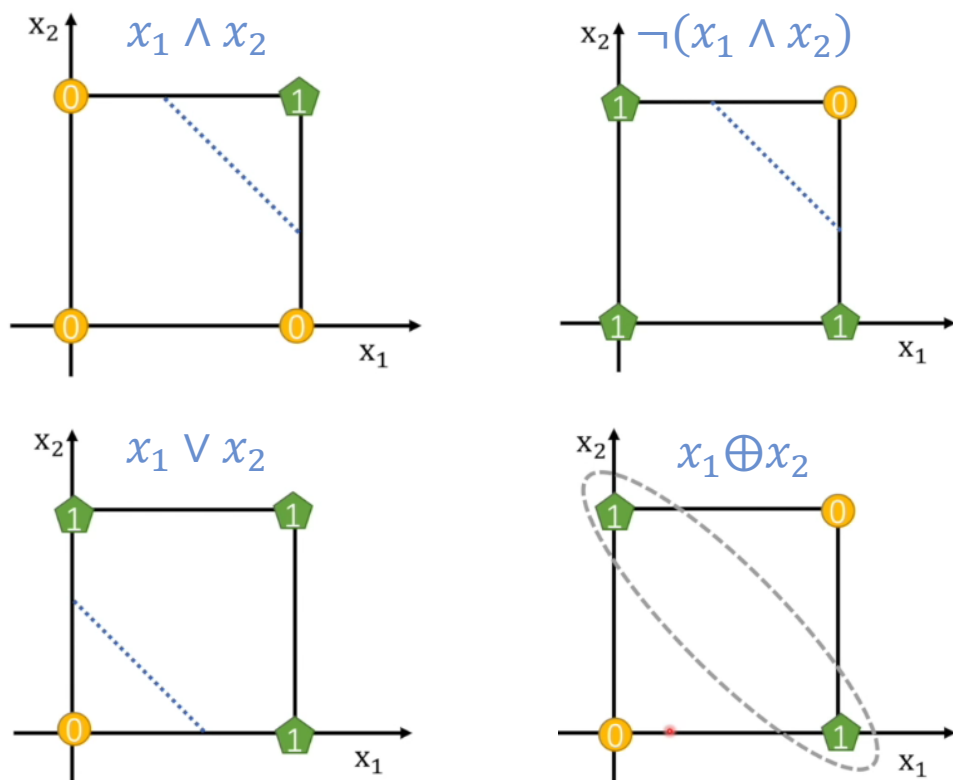


1. <https://blog.csdn.net/xholes/article/details/78461164>

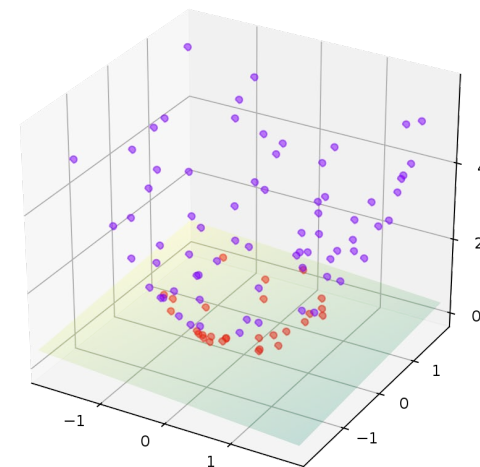
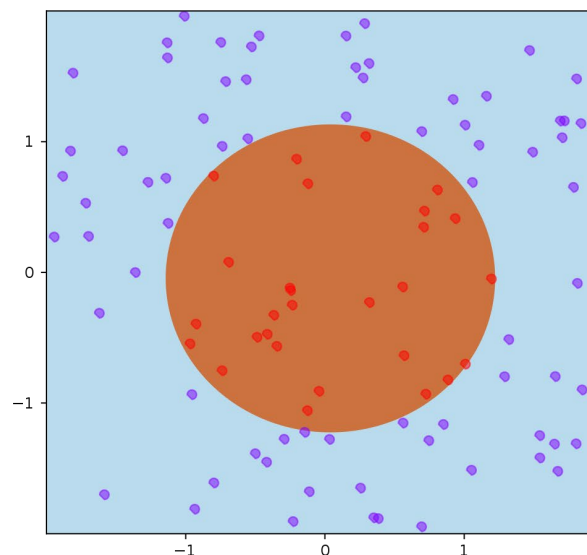
2. <https://www.cnblogs.com/huangyc/p/9706575.html>

1.3 感知机的缺陷

- 无法学习异或函数



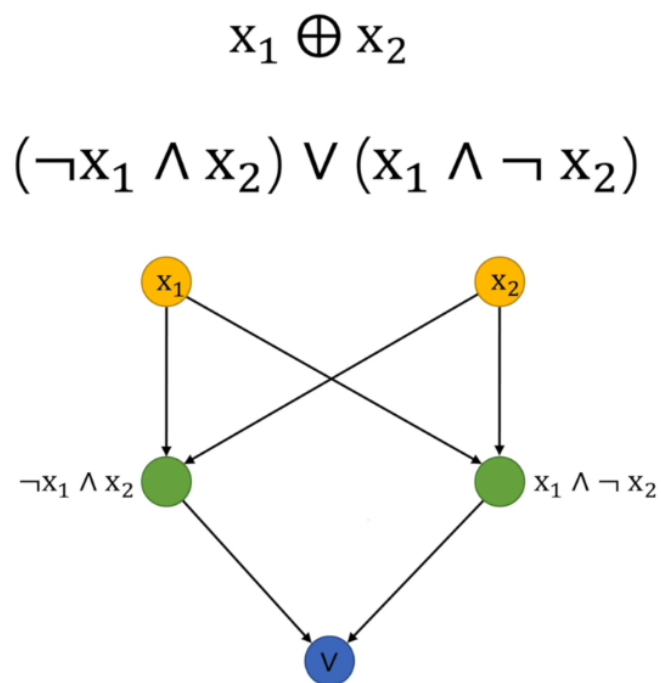
- 难以解决现实中的非线性可分问题



核感知机 (kernel perceptron) 可以将非线性可分的问题转换为高维度线性可分的问题，但是在训练数据集的数据量 m 极大的情况下，每次计算的时间复杂度为 $O(m^2)$ ，显然这是难以接受的。

1.4 多层感知机 (Multilayer Perceptrons)

Q1:如何解决复杂的非线性可分问题?



Q2:如何解决复杂的非线性可分问题?

$$H = XW^{(1)} + b^{(1)}$$
$$O = HW^{(2)} + b^{(2)} \Rightarrow$$

$$O = (XW^{(1)} + b^{(1)})W^{(2)} + b^{(2)}$$
$$= XW^{(1)}W^{(2)} + b^{(1)}W^{(2)} + b^{(2)}$$
$$= XW + b$$

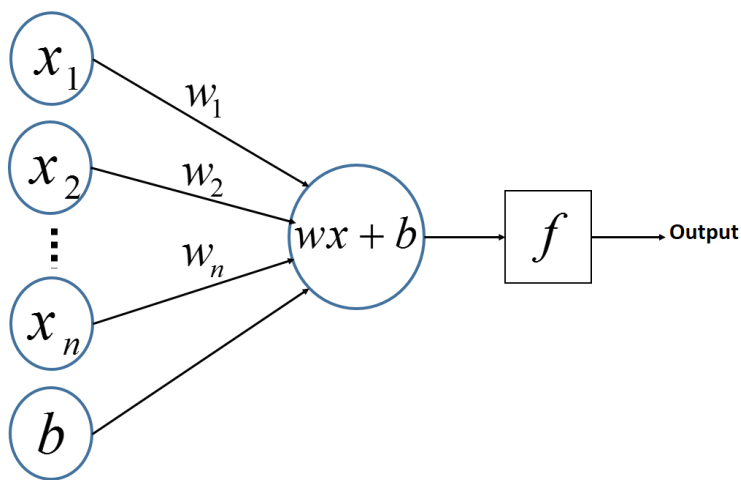
多层线性模型的组合仍然是线性模型!

引入非线性部分→**激活函数**

$$H = \sigma(XW^{(1)} + b^{(1)})$$
$$O = HW^{(2)} + b^{(2)}$$

1.4 多层感知机 (Multilayer Perceptrons)

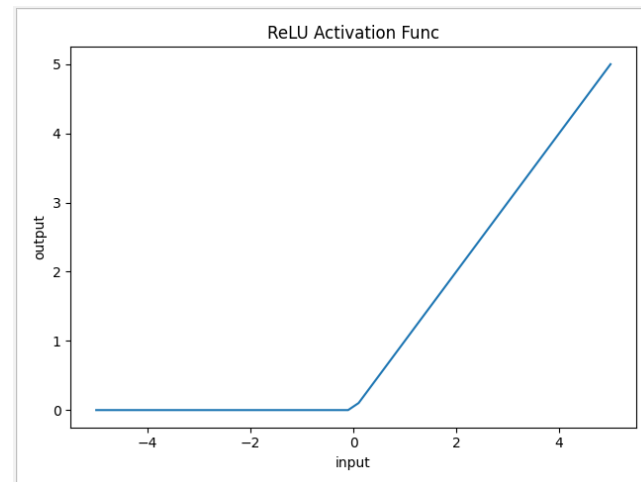
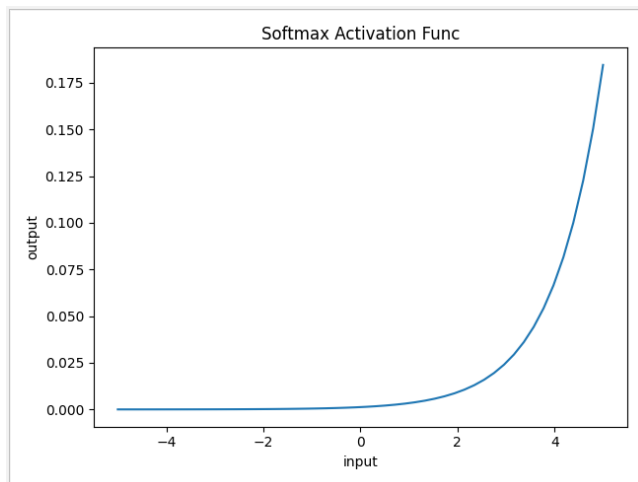
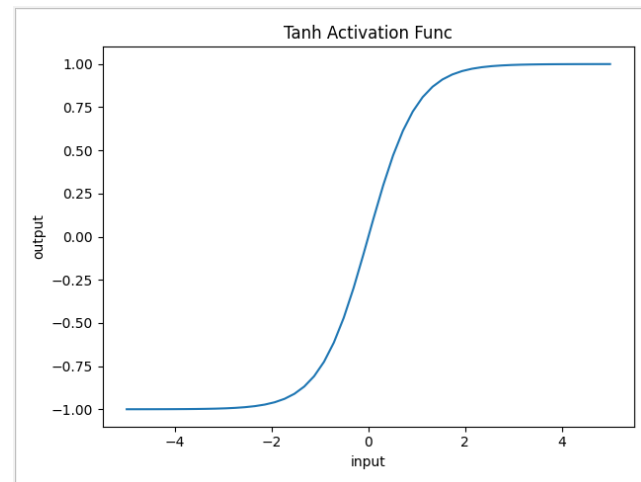
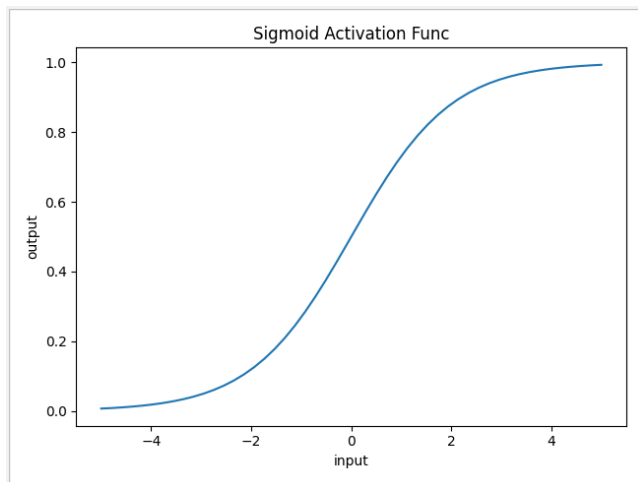
激活函数



感知机的输出如下:

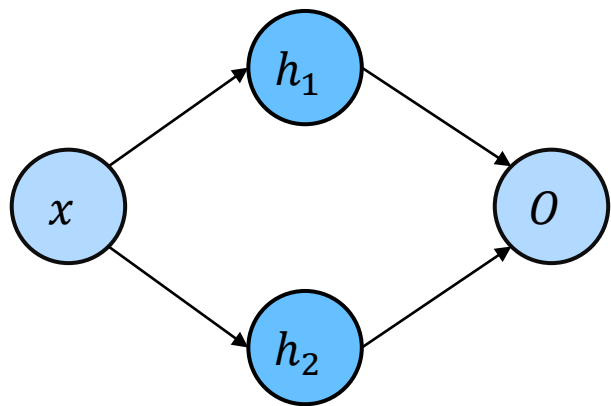
$$y = f(\mathbf{w}^T \mathbf{x} + b) \quad (1-8)$$

$$f(z) = \begin{cases} 0, & z < 0 \\ 1, & z \geq 0 \end{cases} \quad (1-9)$$



1.4 多层感知机 (Multilayer Perceptrons)

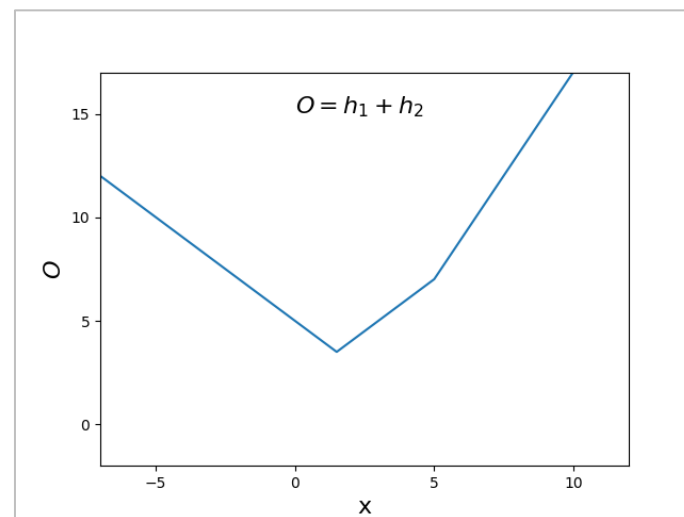
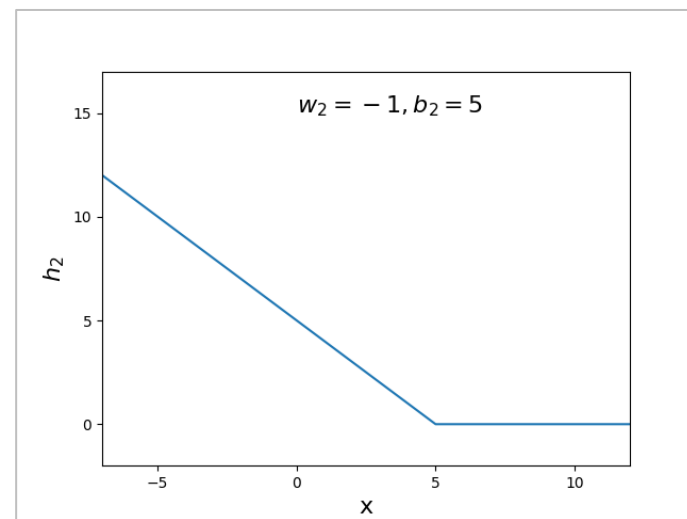
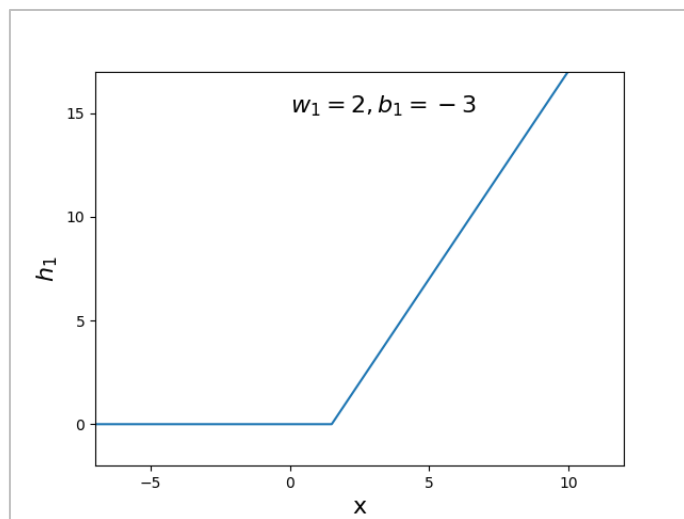
以ReLU为例演示多层感知机进行一维分类时的运行原理



$$h_1 = \text{ReLU}(w_1 * x + b_1)$$

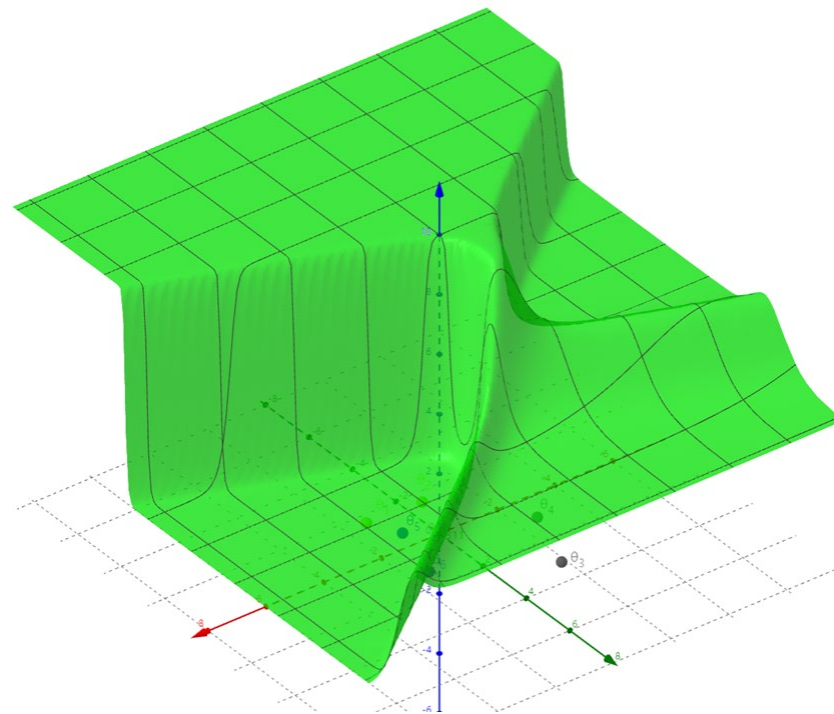
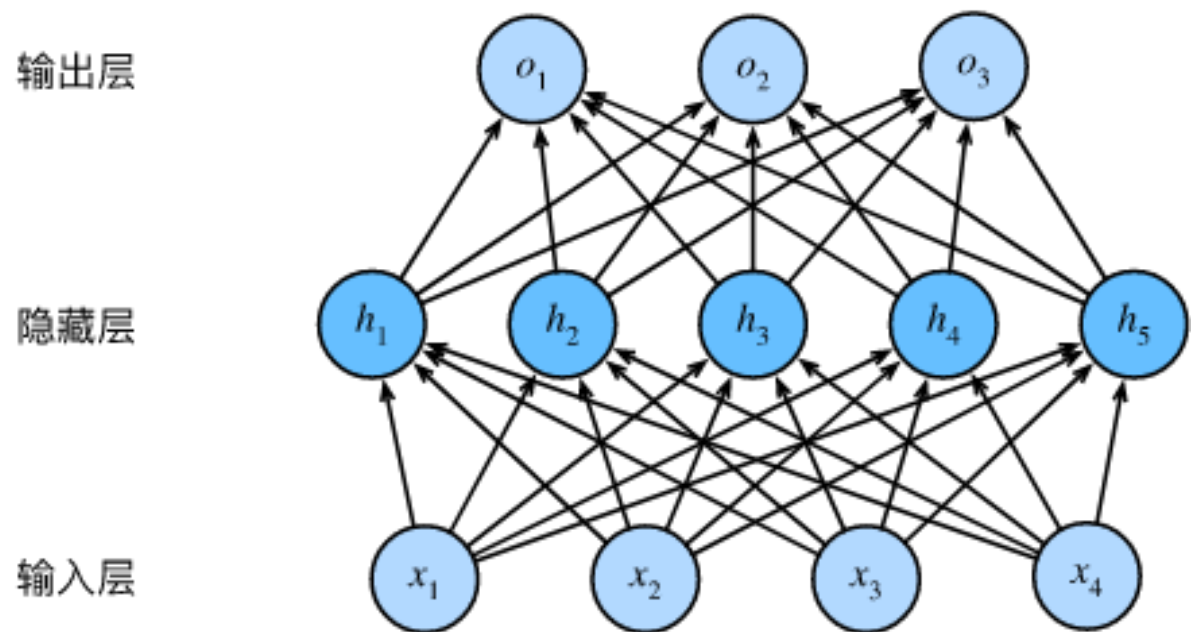
$$h_2 = \text{ReLU}(w_2 * x + b_2)$$

$$O = h_1 + h_2$$

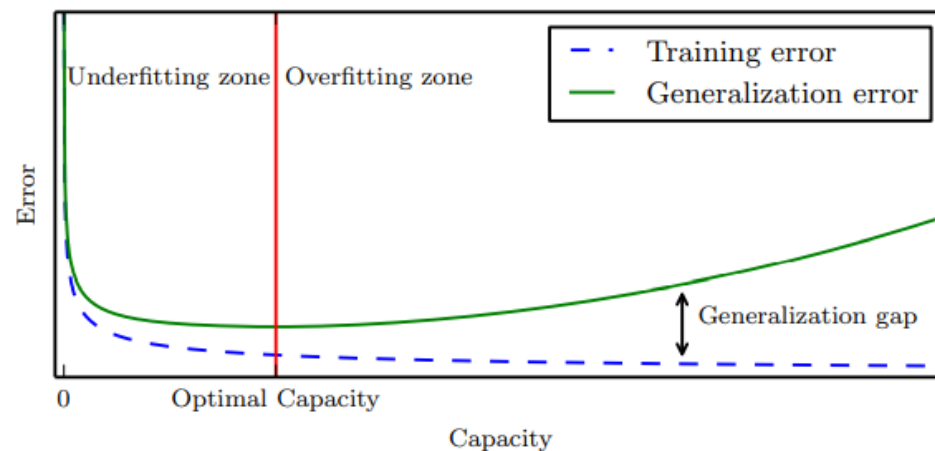
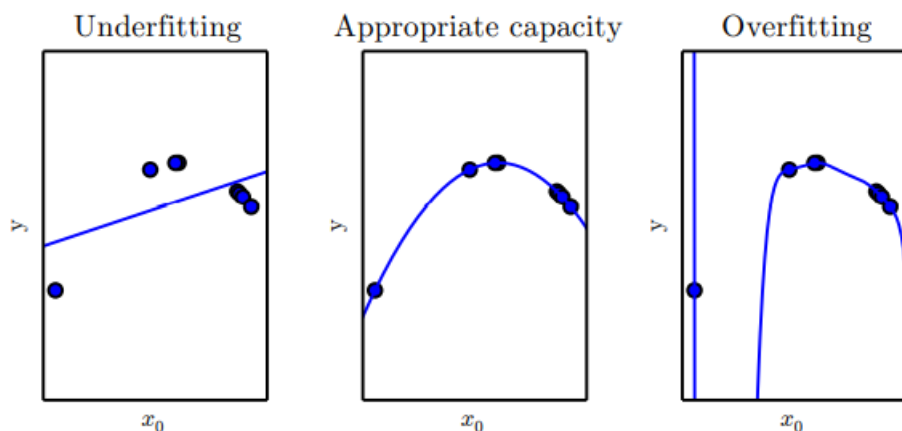


1.4 多层感知机 (Multilayer Perceptrons)

我们可以通过在网络中加入一个或多个隐藏层，使其能处理更普遍的函数关系类型。要做到这一点，最简单的方法是将许多全连接层堆叠在一起。每一层都输出到上面的层，直到生成最后的输出。这种架构称为**多层感知机** (multilayer perceptron)，通常缩写为**MLP**。



1.5 欠拟合和过拟合



过拟合是由于违背独立同分布假设而引起的问题。比如，在一个猫狗的二分类问题中，由于数据集没有黄猫和黑狗，模型不仅学习到了区分猫和狗的必要特征，还学习到了 **“黄毛不是猫，黑毛不是狗”** 的错误知识！

解决方法→正则化：权重衰减、Dropout等



CONTENTS

01

深度学习

02

深度学习优化算法

03

卷积神经网络

04

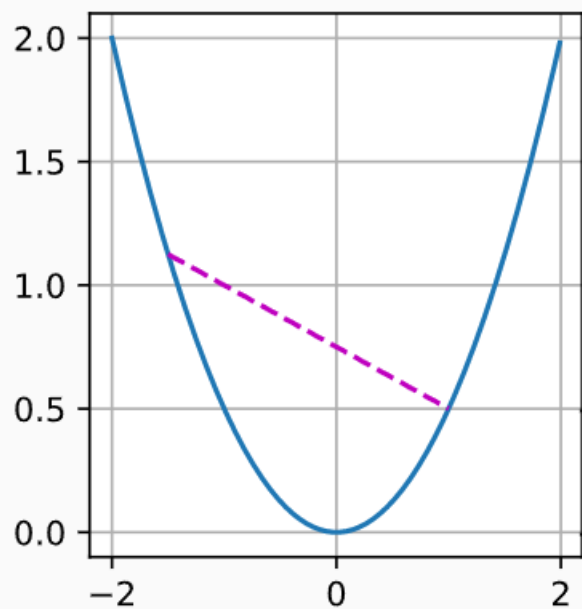
点云数据处理

05

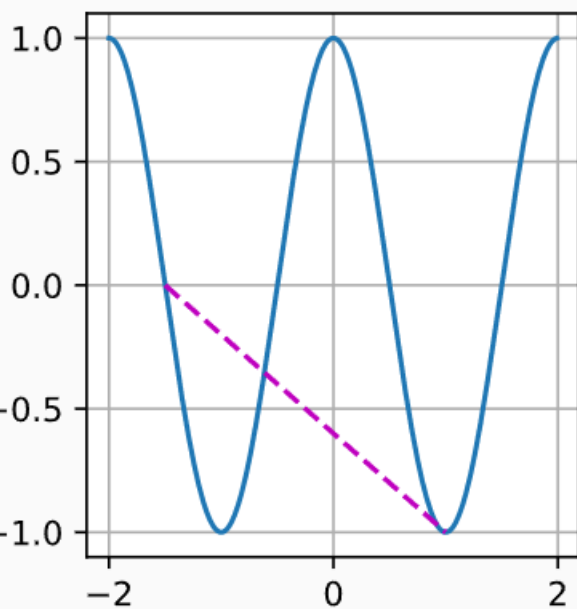
深度学习在点云中的应用

2.1 凸问题

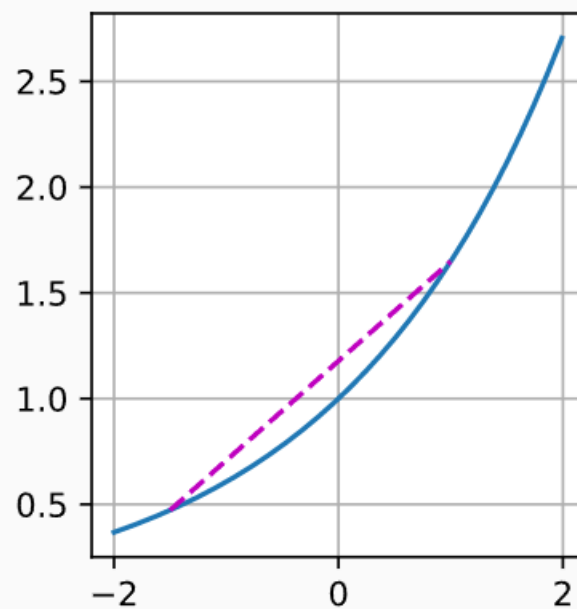
抛物线



余弦函数

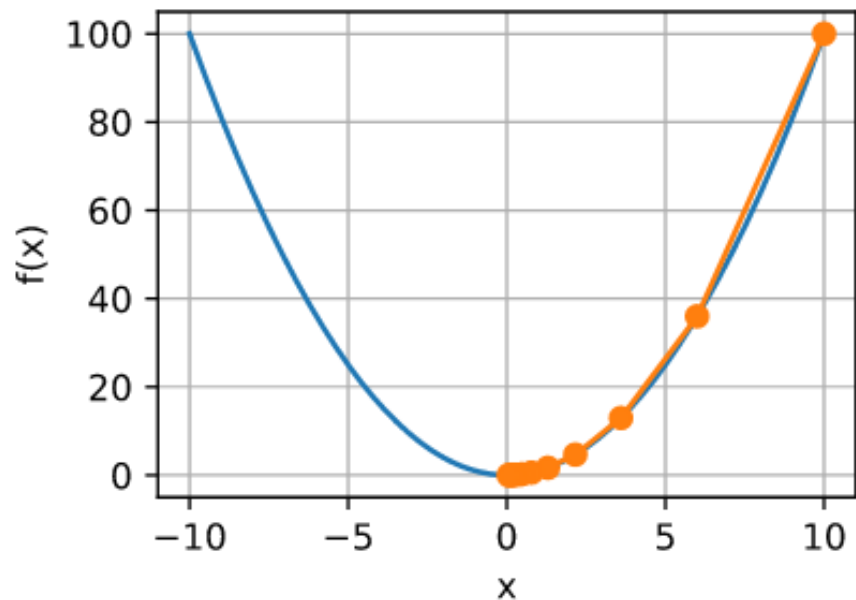


指数函数

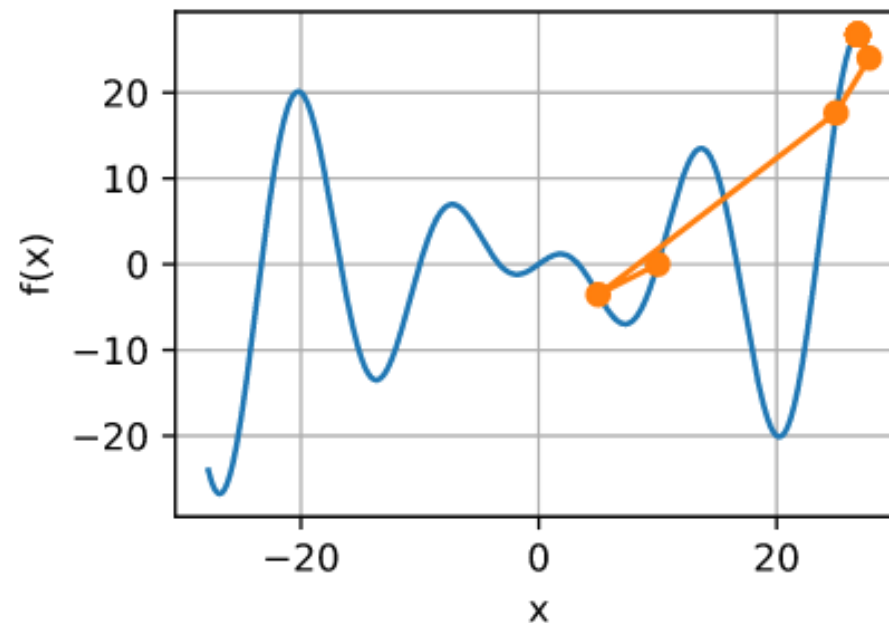


显然，余弦函数是非凸的，而抛物线和指数函数均为凸函数

2.2 优化算法



对于凸函数而言，局部极小值同时也是全局极小值

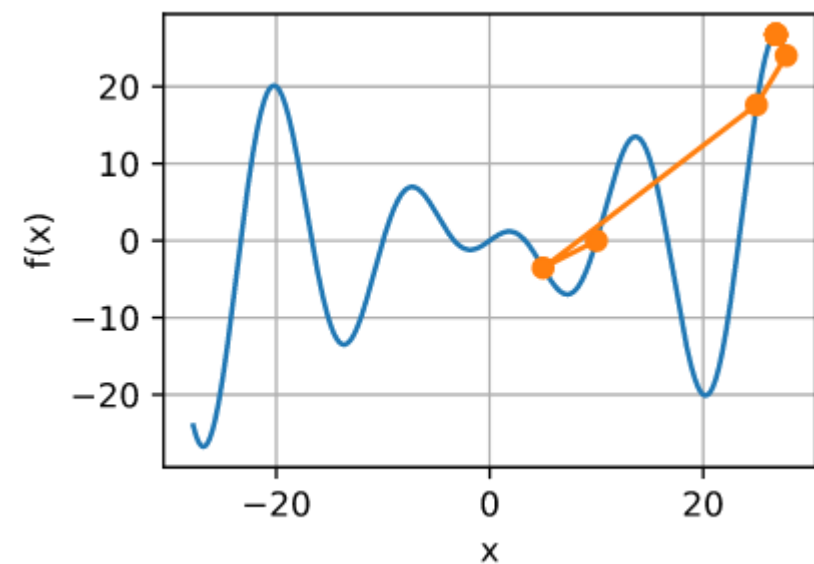


非凸函数存在多个极小值点，找到全局极小值非常困难！

优化算法：随机梯度下降法（SGD）、动量法、AdaGrad、Adam等，这些算法通过优化更新参数的方式，使得算法更可能收敛到较好结果。

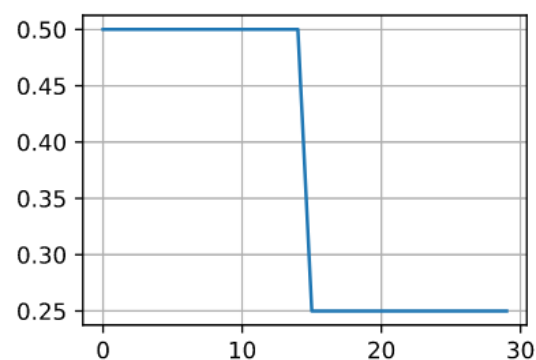
2.3 学习率

学习率过大，优化就会发散；
学习率太小，训练就会需要过长时间

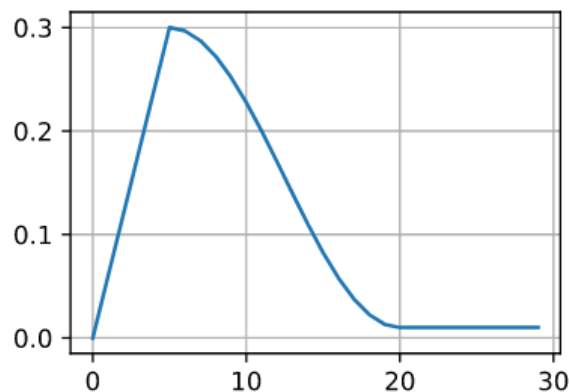
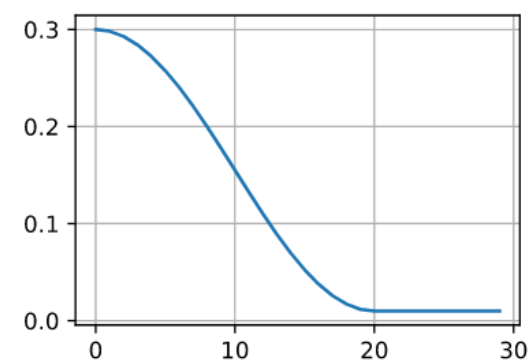


$$\mathbf{w} \leftarrow \mathbf{w} - \frac{\eta}{|\mathbf{B}|} \sum_{i \in \mathbf{B}} \partial_{\mathbf{w}} l^{(i)}(\mathbf{w}, b)$$
$$b \leftarrow b - \frac{\eta}{|\mathbf{B}|} \sum_{i \in \mathbf{B}} \partial_b l^{(i)}(\mathbf{w}, b)$$

手动阶跃式调节



余弦调度器



预热（初始设定较高的学习率可能会导致一开始就发散）



CONTENTS

01

深度学习

02

深度学习优化算法

03

卷积神经网络

04

点云数据处理

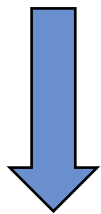
05

深度学习在点云中的应用

3.1 从MLP到卷积

如果直接将图像展平为一个一维向量，再利用MLP处理，那么所需要的参数数量将是无法接受的。

1	1	0
1	0	1
0	0	1



1	1	0	1	0	1	0	0	1
---	---	---	---	---	---	---	---	---

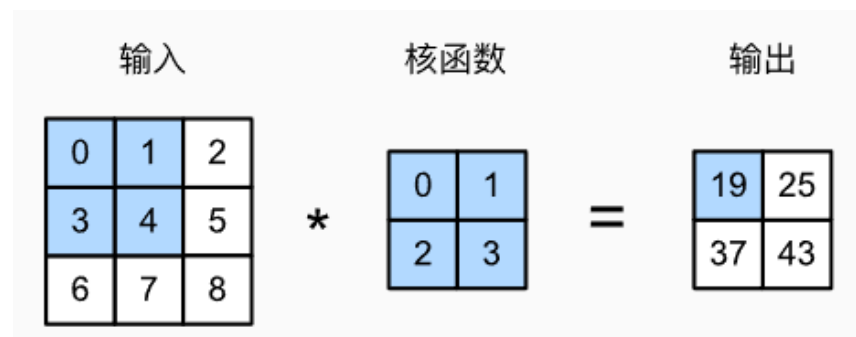
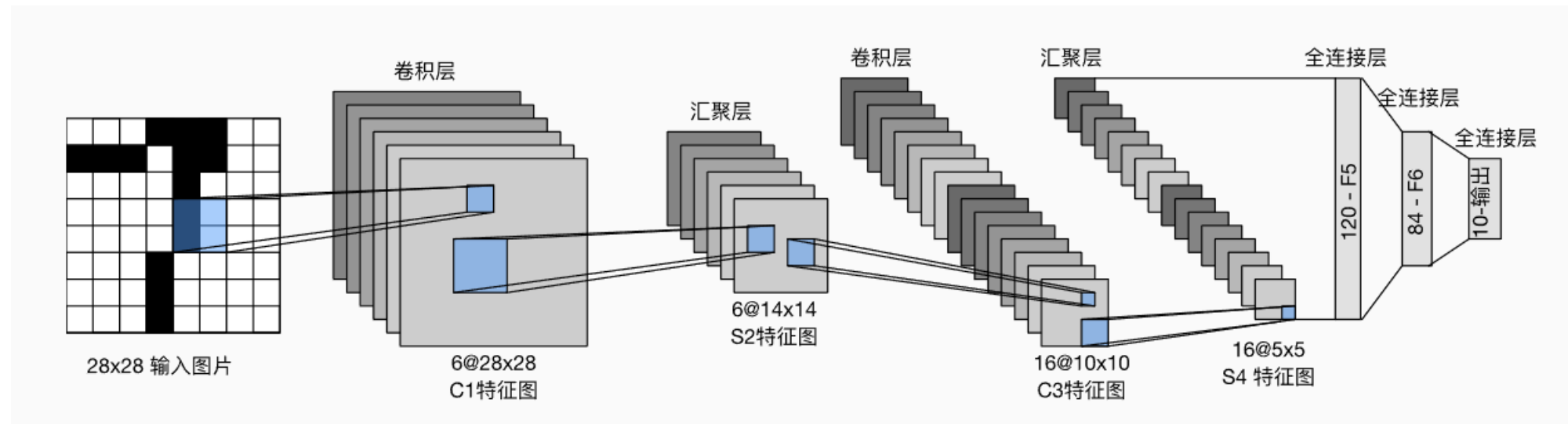


图像特征的性质：

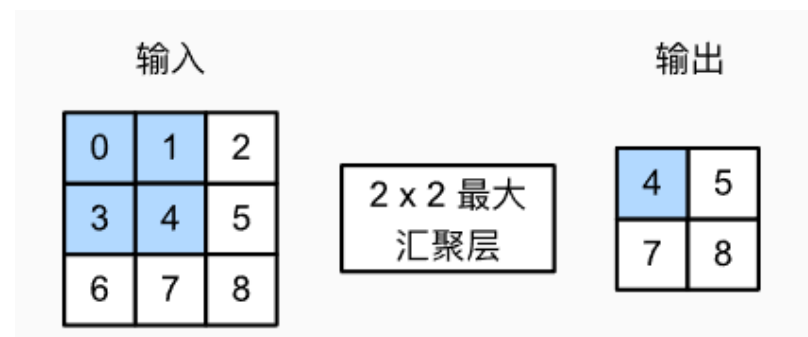
1. 平移不变性 (translation invariance)
2. 局部性 (locality)

3.2 卷积神经网络的基本结构

LeNet
(1994)



卷积层



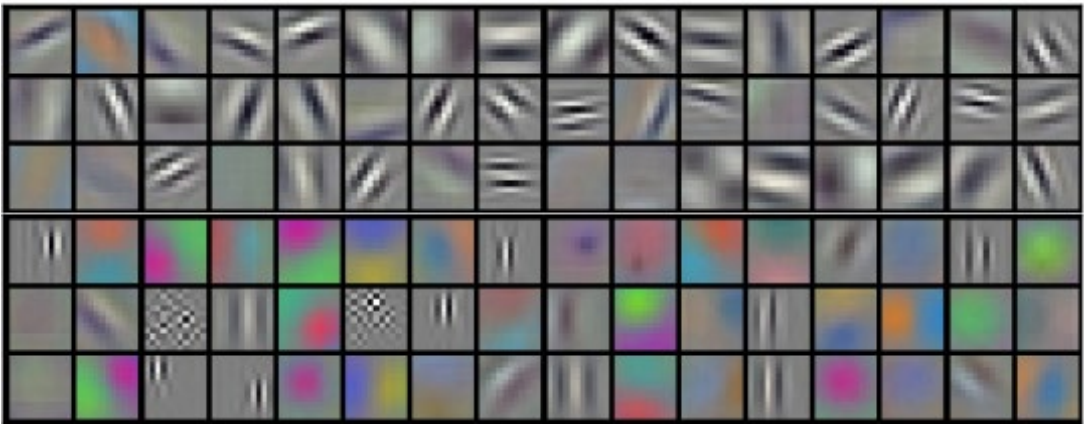
汇聚层 (pooling)

3.3 AlexNet



从LeNet(左)到AlexNet (右)

2012年，AlexNet横空出世。它首次证明了学习到的特征可以超越手工设计的特征吗，一举打破了计算机视觉研究的现状。 AlexNet使用了8层卷积神经网络，并以很大的优势赢得了2012年ImageNet图像识别挑战赛。



AlexNet第一层卷积层学习到的特征



CONTENTS

01

深度学习

02

深度学习优化算法

03

卷积神经网络

04

点云数据处理

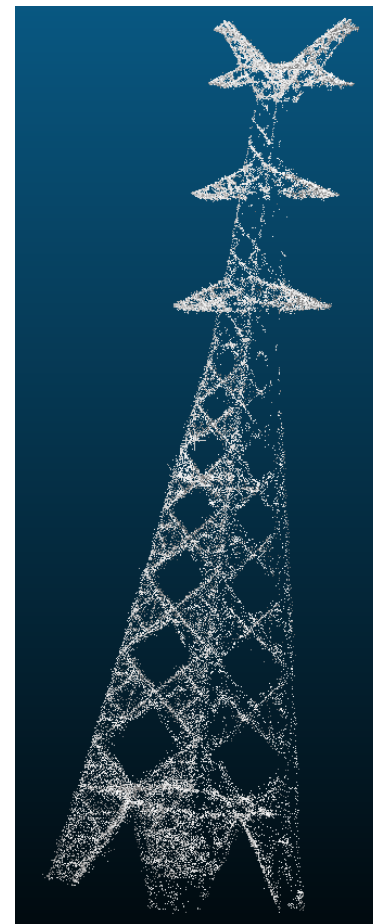
05

深度学习在点云中的应用

4.1 什么是点云?

```
601916.90760040 2409916.66970062 32.09039967 230 233 238
601916.89900208 2409916.60889816 32.17090027 229 230 235
601916.93039703 2409916.56330109 32.11040012 218 227 236
601917.16169739 2409915.70389938 32.50990002 216 226 233
601917.23020172 2409915.59870148 32.47230045 238 245 253
601917.27300262 2409915.54679871 32.43949978 242 244 246
601917.30860138 2409915.47019958 32.39000027 199 196 203
601917.37159729 2409915.38349915 32.34940044 143 132 135
601919.90319824 2409917.55170059 25.72219984 153 159 167
601921.02220154 2409918.35070038 23.02980010 134 146 160
601921.08550262 2409918.10129929 23.06219999 237 249 255
601917.72579956 2409914.28089905 32.63329975 114 129 118
601917.75859833 2409914.15779877 32.71100037 124 124 130
601917.78820038 2409914.03110123 32.83079998 135 125 133
601917.83560181 2409913.87570190 32.90099995 225 231 226
601917.82489777 2409913.83980179 32.92809956 247 255 255
601917.83380127 2409913.70339966 33.05980007 249 248 254
601917.88899994 2409913.58309937 33.00030033 115 124 139
601917.95290375 2409913.51459885 32.94980042 215 220 223
```

杆塔的XYZ点云数据



点云数据常见格式有OFF、PLY、PTS、PTX、XYZ、LAS/LAZ等。其中LAS/LAZ格式文件已成为LiDAR数据的工业标准格式。

4.2 点云数据处理

1. 点云滤波

点云滤波的主要方法有：双边滤波、高斯滤波、条件滤波、直通滤波、随机采样一致滤波、VoxelGrid滤波等，这些算法都被封装在了PCL库中。

2. 点云关键点

常见的三维点云关键点提取算法：ISS3D、Harris3D、NARF、SIFT3D，在PCL库均有实现。

3. 特征和特征描述

常用的特征描述算法有：法线和曲率计算、特征值分析、PFH、FPFH、3D Shape Context、Spin Image等。

4. 点云配准

常用算法有两种：NDT&ICP

5. 点云分割与分类

分割：区域生长、Ransac线面提取、NDT-RANSAC、K-Means、Normalize Cut、3D Hough Transform(线面提取)、连通分析

分类：基于点的分类，基于分割的分类，监督分类与非监督分类

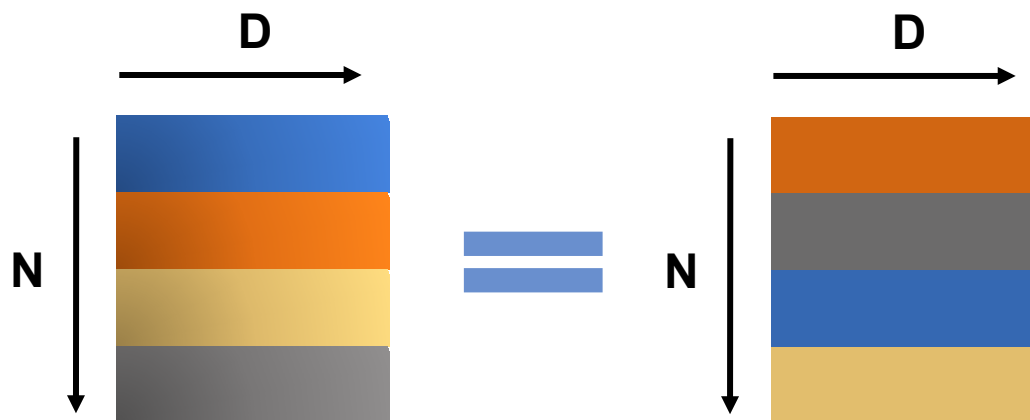
6. 三维重建

泊松重建、Delauary triangulatoins、KinectFusion、Kintinuous、ElasticFusion、ElasticReconstruction、InfiniTAM、DynamicFusion等。

4.3 点云数据的特征

三维点云数据特性：

1. 非结构化数据
2. 无序性排列
3. 旋转不变性
4. 数据点近密远疏
5. 数据缺失



传统点云处理算法需要针对特定的点云数据手动进行特征提取，效率不高，算法鲁棒性也不够好，难以即时高效处理大量点云数据；同时，这些方法也未能充分利用特征之间的一致性和互补信息，因此很难捕获高级语义结构。



CONTENTS

01

深度学习

02

深度学习优化算法

03

卷积神经网络

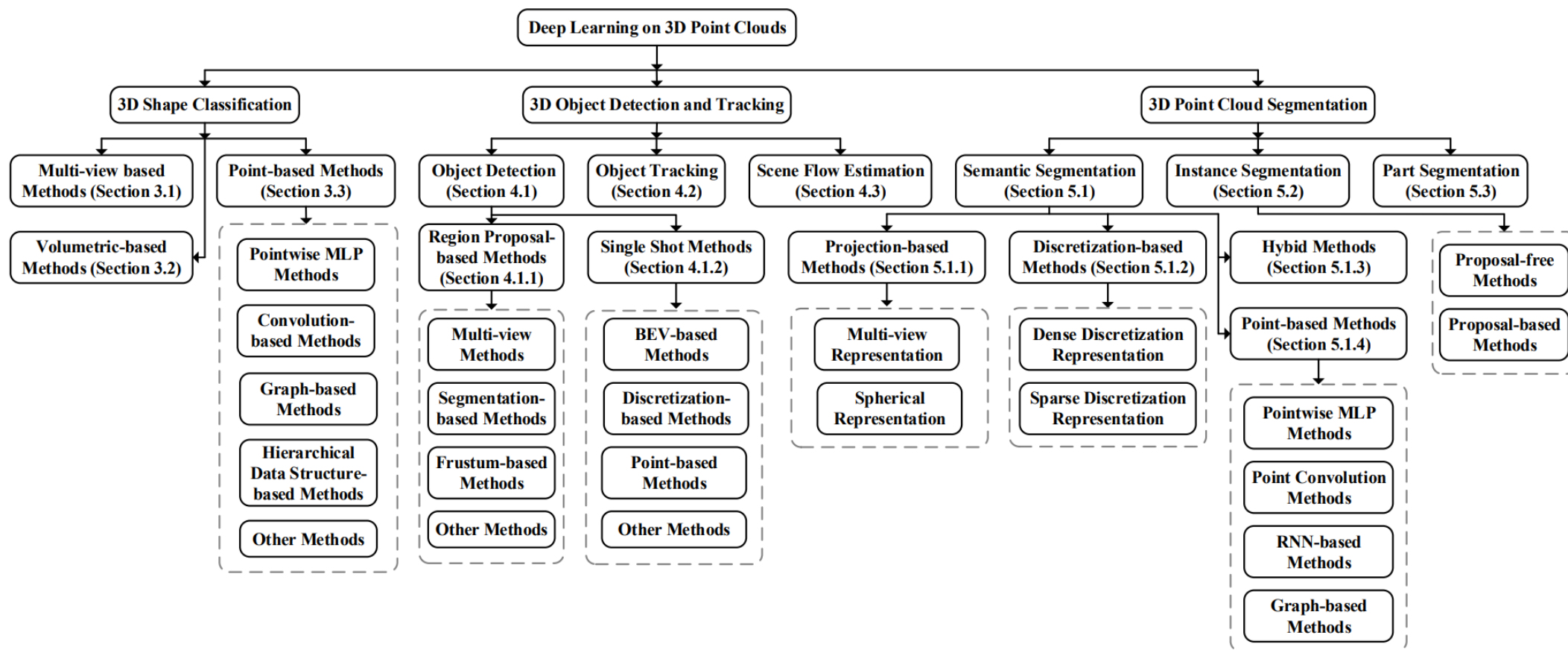
04

点云数据处理

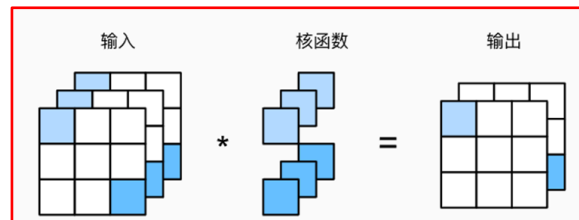
05

深度学习在点云中的应用

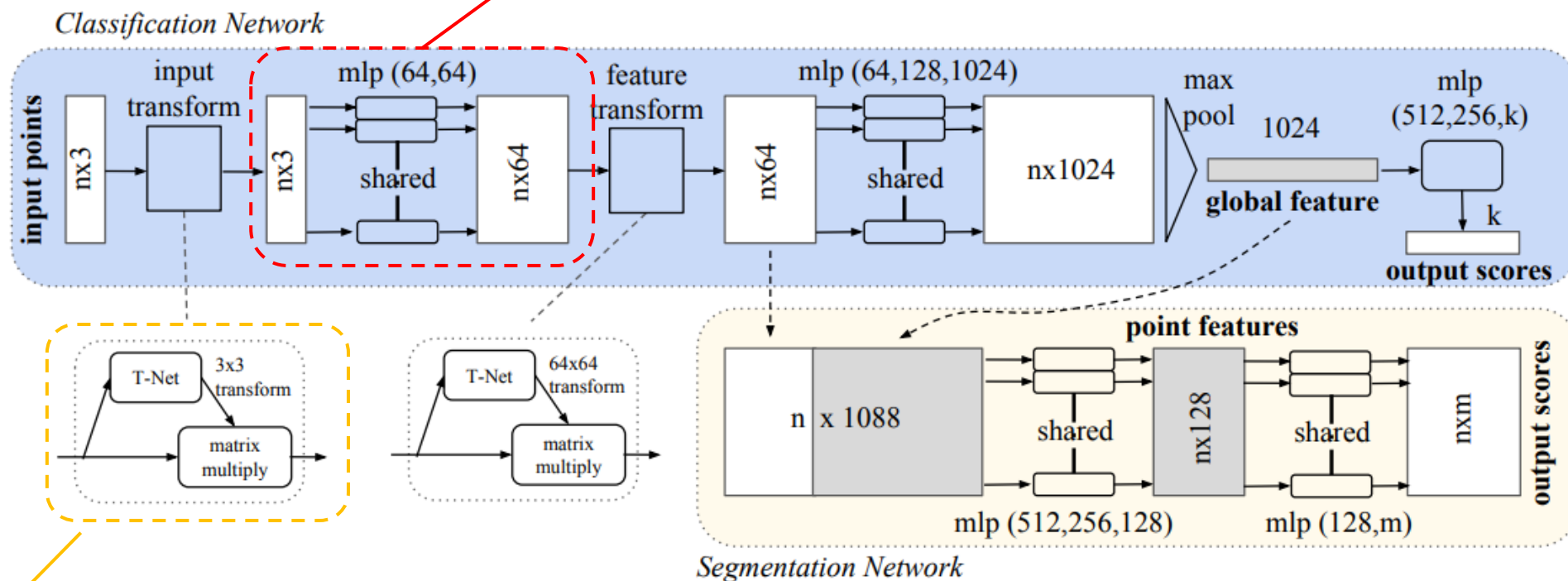
5.1 深度学习在点云中的应用



5.1 PointNet



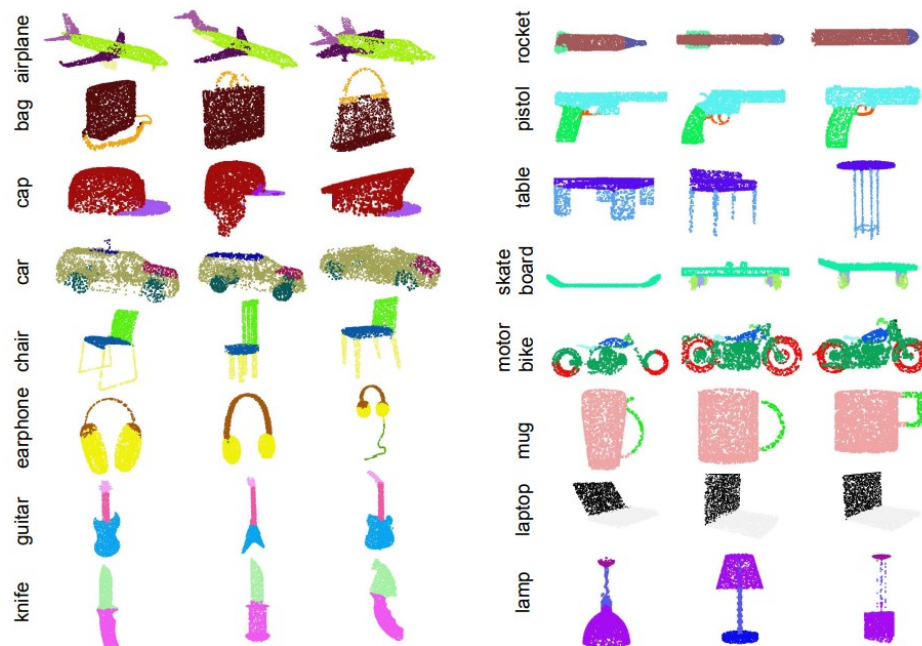
1×1卷积层



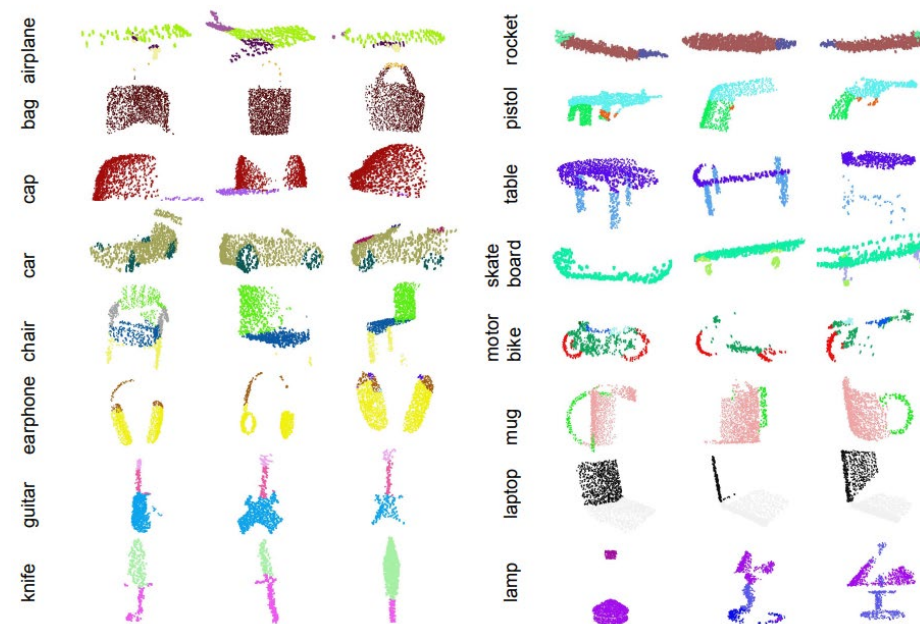
旋转校正

PointNet当时不论是分割还是分类的结果都超过了当时的体素系列网络，同时由于参数少等特点，训练快，属于轻量级网络。

5.1 PointNet



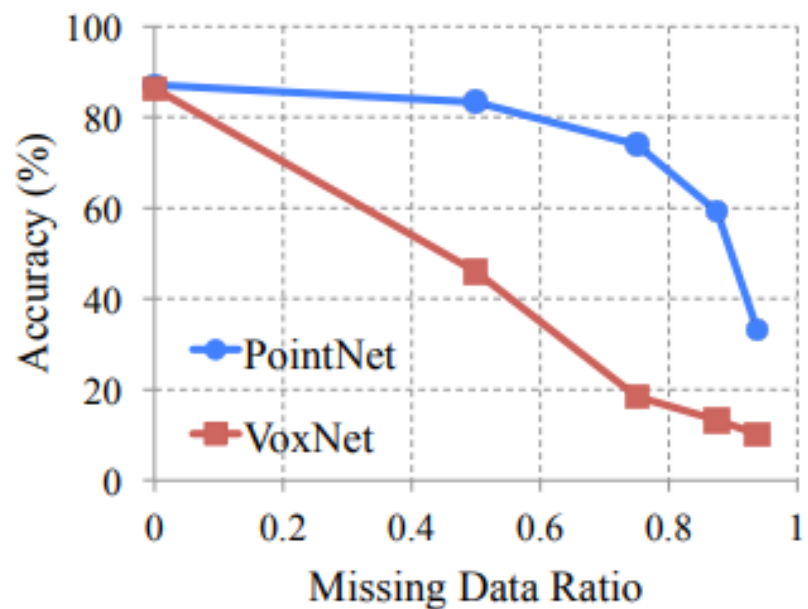
PointNet segmentation results on complete CAD models



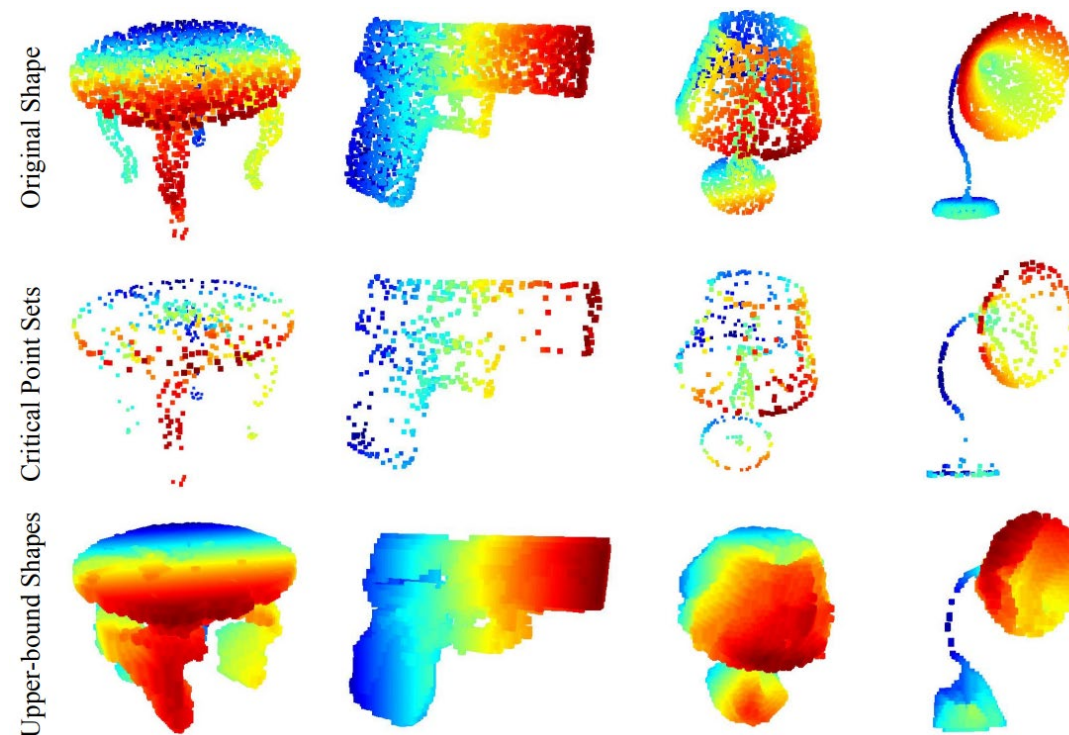
PointNet segmentation results on simulated Kinect scans

5.1 PointNet

算法鲁棒性



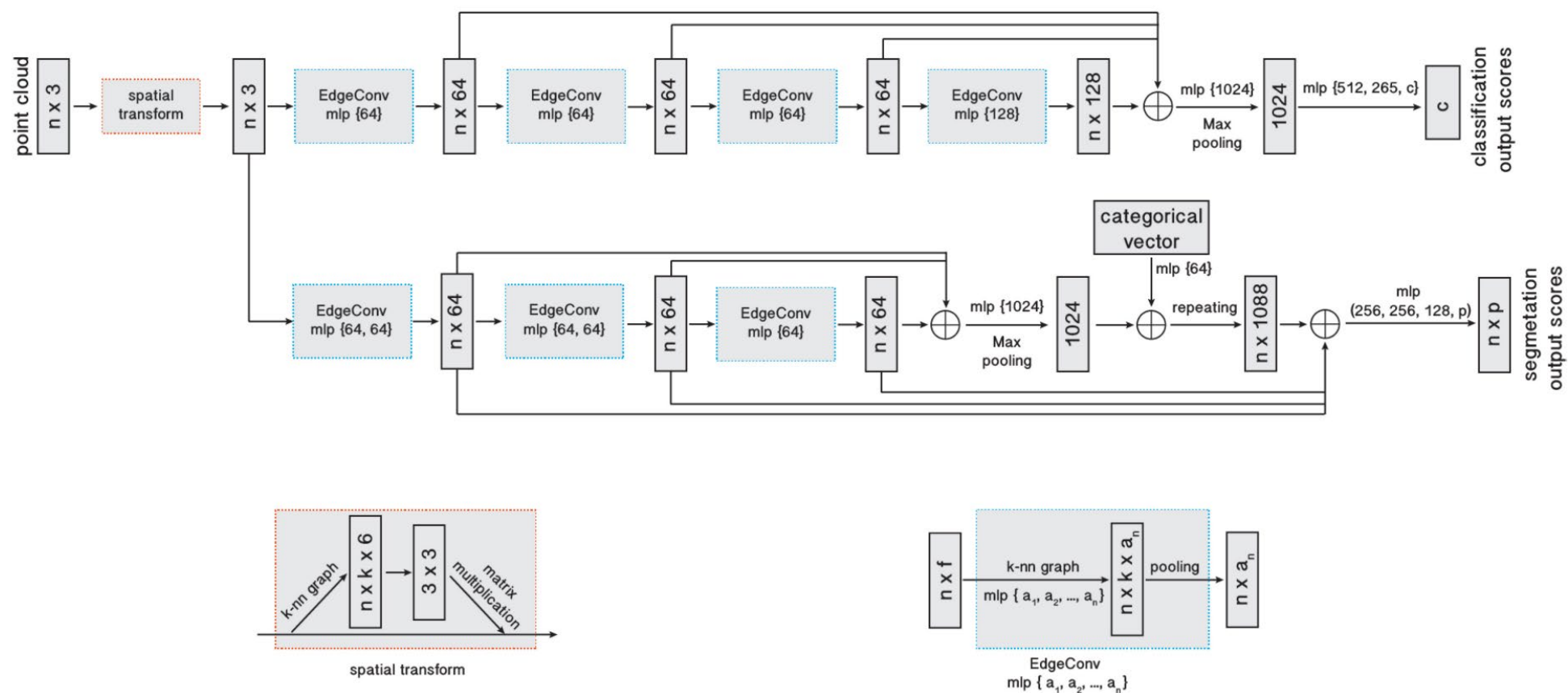
在点云数据缺失60%时，准确率都没有明显降低



能够最大程度激活网络的点都是物体的主干点，将其上采样，很容易就能得到原始的结构。

5.2 DGCNN

动态图神经网络



5.3 PCT

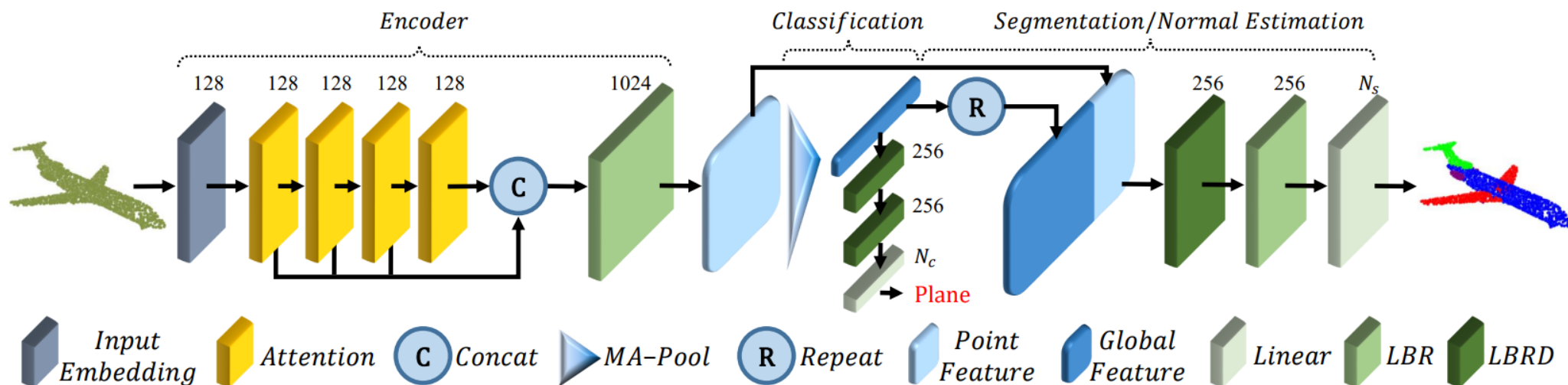


Figure 2. PCT architecture. The encoder mainly comprises an *Input Embedding* module and four stacked *Attention* module. The decoder mainly comprises multiple *Linear* layers. Numbers above each module indicate its output channels. *MA-Pool* concatenates *Max-Pool* and *Average-Pool*. *LBR* combines *Linear*, *BatchNorm* and *ReLU* layers. *LBRD* means *LBR* followed by a *Dropout* layer.

将Transformer迁移到点云学习，也取得了比较好的效果。

参考文献

- [1] 周志华, 机器学习, 清华大学出版社, 2016.
- [2] Goodfellow I, Bengio Y, Courville A. Deep learning[M]. MIT press, 2016.
- [3] Zhang A, Lipton Z C, Li M, et al. Dive into deep learning[J]. arXiv preprint arXiv:2106.11342, 2021.
- [4] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[J]. Advances in neural information processing systems, 2012, 25.
- [5] Guo Y, Wang H, Hu Q, et al. Deep learning for 3d point clouds: A survey[J]. IEEE transactions on pattern analysis and machine intelligence, 2020, 43(12): 4338-4364.
- [6] Qi C R, Su H, Mo K, et al. Pointnet: Deep learning on point sets for 3d classification and segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 652-660.
- [7] Qi C R, Yi L, Su H, et al. Pointnet++: Deep hierarchical feature learning on point sets in a metric space[J]. Advances in neural information processing systems, 2017, 30.
- [8] Wang Y, Sun Y, Liu Z, et al. Dynamic graph cnn for learning on point clouds[J]. Acm Transactions On Graphics (tog), 2019, 38(5): 1-12.
- [9] Guo Y, Wang H, Hu Q, et al. Deep learning for 3d point clouds: A survey[J]. IEEE transactions on pattern analysis and machine intelligence, 2020, 43(12): 4338-4364.
- [10] Guo M H, Cai J X, Liu Z N, et al. Pct: Point cloud transformer[J]. Computational Visual Media, 2021, 7(2): 187-199.



- **Thank you!**
