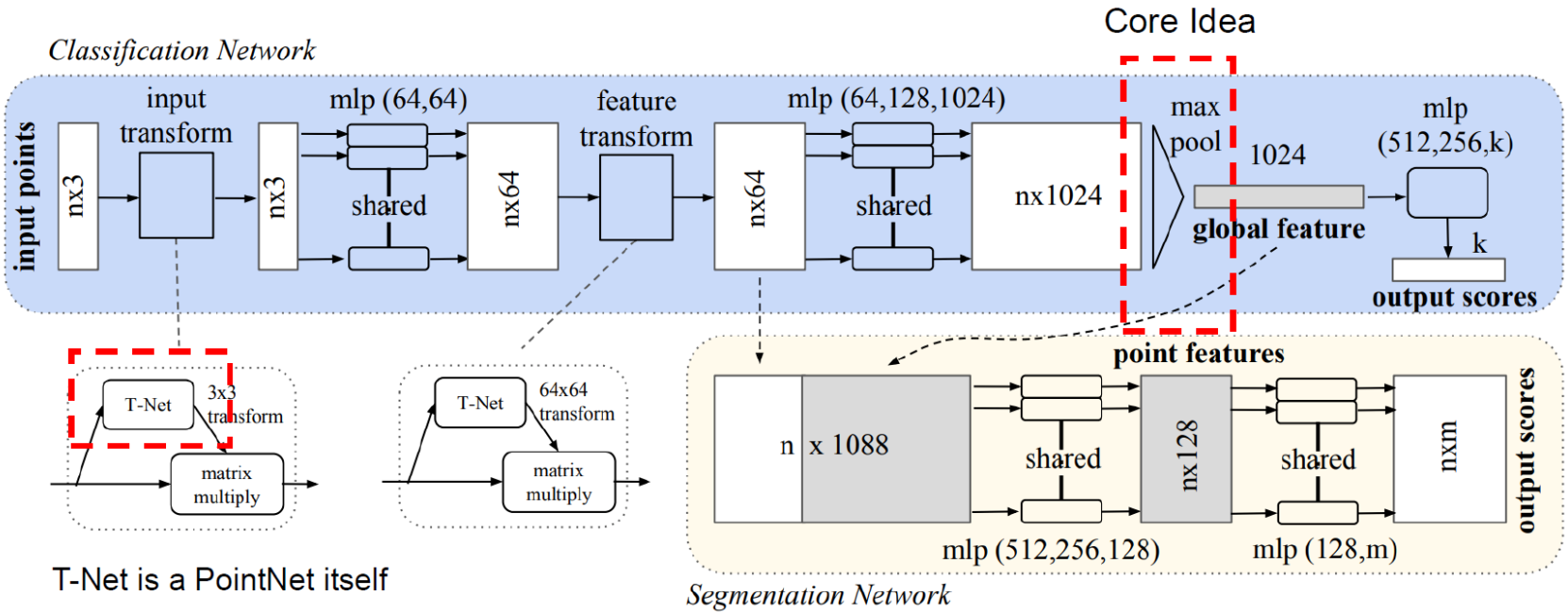
深蓝学院
shenlanxueyuan.com

第五章思路讲解

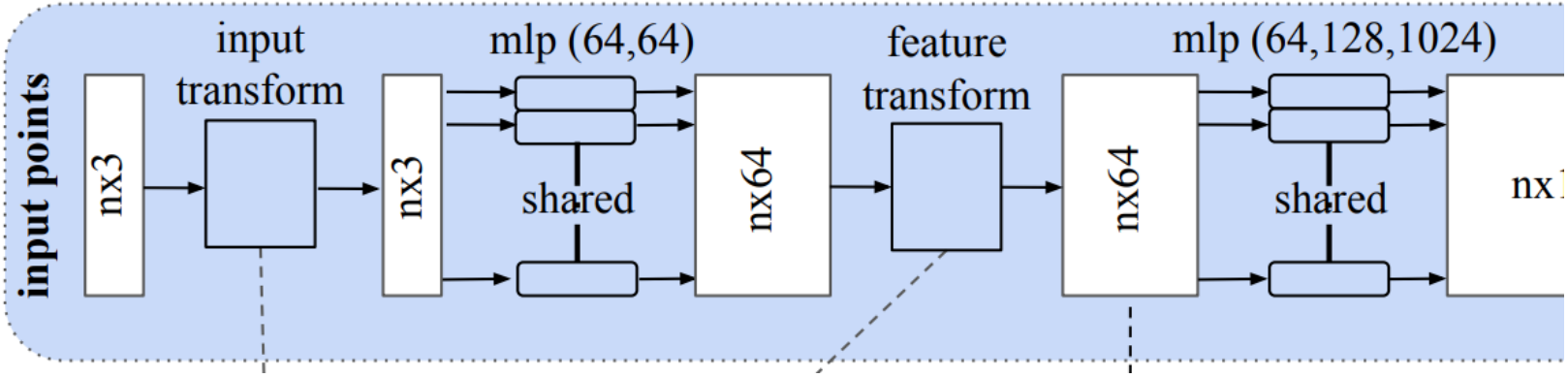
主讲人 王永浩

PointNet



通过分享一版我自己实现的代码（基于pytorch），给大家详细解释一下各部分的实现细节

PointNet




```
self.mlp1 = pt_utils.SharedMLP(mlp1,bn = False)
self.mlp2 = pt_utils.SharedMLP(mlp2,bn = False)
```

（没有T-Net和feature transform结构）

该部分要点：SharedMLP的实现

PointNet

深蓝学院 shenlanxueyuan.com

Pytorch版本：使用conv1D()函数来实现SharedMLP的操作，注意输入和输出的通道 $(N, C_{in}, L) \rightarrow (N, C_{out}, L)$[参考Pytorch官方doc]

In the simplest case, the output value of the layer with input size (N, C_{in}, L) and output (N, C_{out}, L_{out}) can be precisely described as:

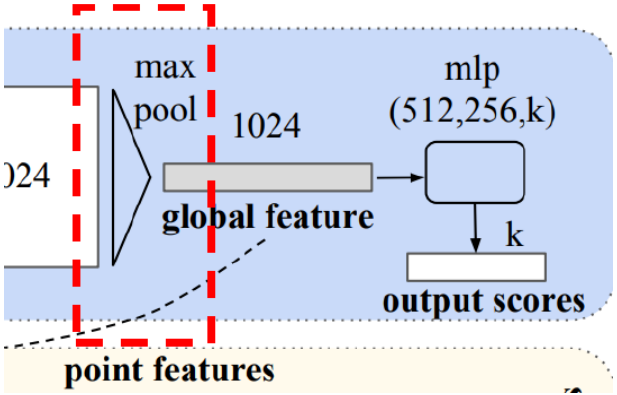
$$\text{out}(N_i, C_{out_j}) = \text{bias}(C_{out_j}) + \sum_{k=0}^{C_{in}-1} \text{weight}(C_{out_j}, k) \star \text{input}(N_i, k)$$

SharedMLP结构

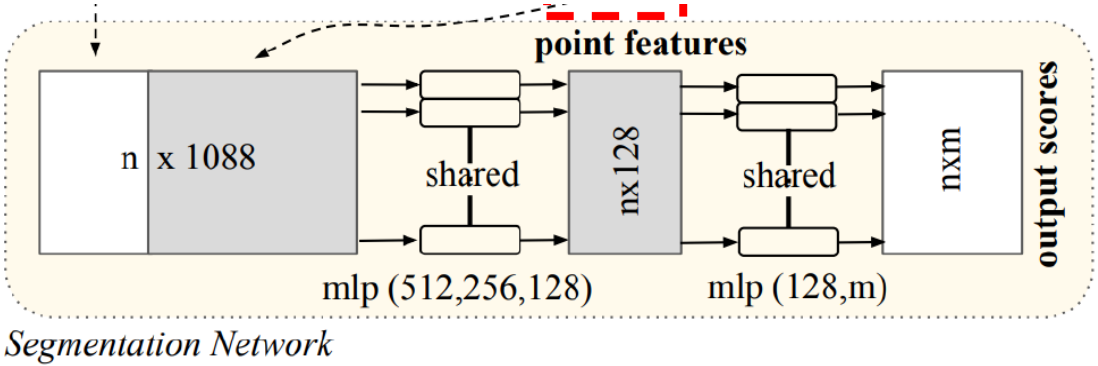
```python
class SharedMLP(nn.Sequential):

    def __init__(
            self,
            args: List[int],
            *,
            bn: bool = False,
            activation=nn.ReLU(inplace=True),
            preact: bool = False,
            first: bool = False,
            name: str = "",
            instance_norm: bool = False,
    ):
        super().__init__()

        for i in range(len(args) - 1):
            self.add_module(
                name + 'layer{}'.format(i),
                Conv1d(
                    args[i],
                    args[i + 1],
                    bn=(not first or not preact or (i != 0)) and bn,
                    activation=activation
                    if (not first or not preact or (i != 0)) else None,
                    preact=preact,
                    instance_norm=instance_norm
                )
            )
```

PointNet

Core Idea



Classification head



Segmentation Network

Segmentation head

分类部分和分割部分

PointNet


```
if segmentaion:
    seg_layers = []
    for k in range(0,tailmlp.__len__()-2):
        seg_layers.append(pt_utils.Conv1d(
            tailmlp[k],
            tailmlp[k+1],
            bn = True
        ))
    seg_layers.append(pt_utils.Conv1d(tailmlp[-2],tailmlp[-1],activation=None))
    self.segmlp = nn.Sequential(*seg_layers)
else:
    cls_layers = []
    for k in range(0,tailmlp.__len__()-2):
        cls_layers.append(pt_utils.Conv1d(
            tailmlp[k],
            tailmlp[k+1],
            bn = True
        ))
    cls_layers.append(pt_utils.Conv1d(tailmlp[-2],tailmlp[-1],activation=None))
    self.clsmlp = nn.Sequential(*cls_layers)
```

PointNet

Forward部分

```
def forward(self,x):
    n_pts = x.size()[1]
    if self.useforseg:
        x = self.mlp1(x)
        gl_feature = self.mlp2(x)
        gl_feature = torch.max(gl_feature,2,keepdim=True)[0]
        gl_feature = gl_feature.view(-1,1024,1).repeat(1,1,n_pts)
        gl_feature = torch.cat([x,gl_feature],1)
        gl_feature = self.segmlp(gl_feature)
    else:
        x = self.mlp1(x)
        x = self.mlp2(x)
        gl_feature = torch.max(x,2,keepdim=True)[0]
        gl_feature = self.clsmlp(gl_feature)

    return gl_feature
```

这里的代码对照PointNet结构图就可以理解

Dataloader

深蓝学院
shenlanxueyuan.com

数据准备是机器学习中重要的一部分，Dataloader是pytorch提供的有效的数据加载工具，同时还可以整合数据增强的代码，使代码结构更加简洁。

```
def __getitem__(self, idx):

    if torch.is_tensor(idx):
        idx = idx.tolist()
    sample_data = self.data[idx]
    if self.train:
        sample_data = self.transform(sample_data)

    sample = {'pts_input':sample_data,'cls_labels':np.array(self.label[idx])}

    return sample
```

具体的写法参考Pytorch官方文档

Train and Test

深蓝学院 shenlanxueyuan.com

一般基于Pytorch 的Train和Test都是比较固定的写法。

```
def train_and_eval(model, train_data, eval_data, tb_log, ckpt_dir, log_f, last_epoch = -1):

    model.cuda() # 将模型迁移到GPU上
    optimizer = optim.Adam(model.parameters(), lr=args.lr, weight_decay=args.weight_decay) #设定优化器

    lr_scheduler = torch.optim.lr_scheduler.MultiStepLR(optimizer, milestones = args.decay_step_list, gamma=args.lr_decay, last_epoch = last_epoch)

    total_it = 0
    best_acc = -1e+8
    best_epoch = 0

    for epoch in range(1, args.epochs + 1):

        total_it = train_one_epoch(model, train_data, optimizer, epoch, lr_scheduler, total_it, tb_log, log_f)
```

可以进一步参看code_example中的train_and_eval.py

其他一些学习建议

1. 多看文档
2. 多看论文的优秀源码实现
3. 多动手

在线问答

深蓝学院
shenlanxueyuan.com

Q&A

深蓝学院
shenlanxueyuan.com

感谢各位聆听！

Thanks for Listening