

深度学习模块排列组合教程(pytorch, 以UNet为例)

对应视频教程: [视频地址](#)

大家好, 由于我的视频中没有配音, 很多小伙伴看不懂, 所以我现在加上配音, 希望大家可以有所收获。首先是卷积替换操作, 就是把nn.conv2d替换成卷积模块, 这里我用深度可分离卷积替换nn.conv2d, 首先我先打印了一下模型的结构, 截个图, 看一下替换掉nn.conv2d后有什么区别。我截图的部分就是要替换卷积的部分。这里我已经给出了深度可分离卷积的代码, 把这个类名复制下来, 直接替代nn.conv2d就可以完成卷积替换操作了。大家可以对比一下前后的区别。

1.卷积替换

这里用深度可分离卷积举个例子, 代码如下:

```
class DepthwiseConv2d(nn.Module):
    def __init__(self, in_channels, out_channels, kernel_size=1, stride=1,
padding=0, dilation=1, bias=False):
        super(DepthwiseConv2d, self).__init__()

        self.conv1 = nn.Conv2d(in_channels, in_channels, kernel_size, stride,
padding, dilation, groups=in_channels,
                                bias=bias)
        self.pointwise = nn.Conv2d(in_channels, out_channels, 1, 1, 0, 1, 1,
bias=bias)

    def forward(self, x):
        x = self.conv1(x)
        x = self.pointwise(x)
        return x
```

替换方法为:

```
nn.Conv2d(in_channels, out_channels, kernel_size=3, padding=1) # 替换前
DepthwiseConv2d(in_channels, out_channels, kernel_size=3, padding=1) # 替换后
```

替换后, 对比可以看出, 深度可分离卷积确实是有轻量化这个优点。(上图为替换前, 下图为替换后)

UNet total parameters: 13.40M (13395394)

UNet total parameters: 1.54M (1538971)

2.加注意力模块

第二个是往网络里加注意力模块, 其实大家遇到最多的问题就是通道数无法对齐, 所以大家可以像我一样, 一开始就把shape全部打印出来, 这样就可以很好地解决这一问题。这里要注意, 不要在forward中实例化注意力模块, 不然训练起来非常非常慢。像视频中一样, 在forward前面实例化注意力模块。在视频中我直接把注意力放在了下采样代码中, 放在了池化操作和卷积操作的后面。大家可以看到, 我

也遇到了通道数不匹配的问题，遇到这样的问题时，我建议大家直接把报错的信息粘贴给gpt，根据gpt提供的信息可以提高效率，发现问题所在。我这里是要求64通道，但实际上我的是128通道，所以我把in channels改成了out channels，解决了通道数不匹配这一问题。下采样操作执行了四次，可以看到，每次下采样都加上了注意力模块。

首先，print出通道数

在示例中，我将MSCAAttention放在了下采样过程中，参数量对比下图所示，(上图为替换前，下图为替换后)

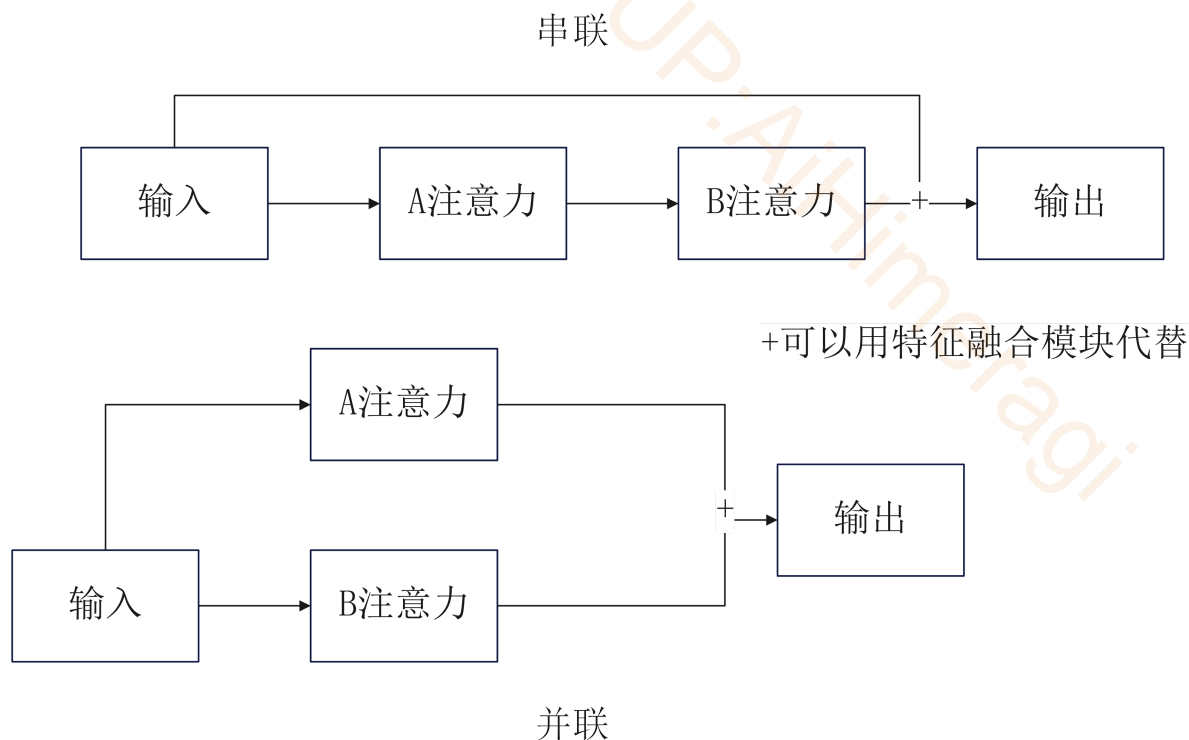
```
UNet total parameters: 1.54M (1538971)
```

```
UNet total parameters: 3.52M (3516699)
```

2.1 注意力模块的排列组合

接下来我想介绍一下模块的排列组合方式，最简单的就是我们初中物理学过的串联和并联两种方式，其中在串联方式中可以添加一个残差连接的操作，就是把输出和原始输入相加，其中相加操作可以用特征融合模块代替。首先我们先来看模块的串联操作，这里我定义了一个类，类名就是串联的拼音，然后呢我把串联的两个注意力模块分别命名为model1和model2，这里要注意传入的参数，在forward中实现了串联的操作，对照着图就很容易理解代码，完成了之后我就用这个模块来代替之前的注意力模块，由于代码不在同一个文件中，这里先把它导入一下，这里可以看到需要传入输入通道数和输出通道数这两个参数，把这两个参数设置好后就可以运行代码了。打印网络结构可以看到咱们已经成功了。

紧接着咱们来看并联操作，和串联操作不同的是，原始输入分别经过两个注意力模块，得到两个输出，这两个输出相加后就是最终的输出，这里我把图截下来，代码和图对照着看就很清晰。写好这个模块之后，咱们再来运行一下，这里报错是因为没有导入并联模块，导入后就可以成功运行代码了。



3.特征融合模块代替cat

第三个是特征融合模块，特征融合模块是拿来代替类似于concat这类函数的操作的，在u net中，就可以在跳跃连接中应用特征融合模块，咱们这里依然是先打印一下模型的输出形状，我这里拿来举例子的特征融合模块只需要传入输入通道数，先按照模型的形状来测试一下这个特征融合模块，可以看到输出和输入的形状是保持一致的，我这里把相加操作改为concat操作，可以看到输出的通道数是翻倍了的，和本例子中的u net模型输出保持了一致，这样才能成功地运行这个代码。可以看到我这里还是报错了，还是通道数不匹配的问题，要求是1024通道，但我只有512通道，既然这样，我就把输入通道数除以二，要求就从1024变成了512，代码就能跑通了。

这里我使用EFF模块来代替cat操作

4. 替换下采样操作

第四个是替换下采样操作，这里可以看到下采样模块需要传入两个参数，输入通道数和输出通道数，咱们先把实例化部分的代码复制粘贴到模型中的下采样部分中，修改输入通道和输出通道这两个参数，把池化操作的代码注释掉，这样就完成了下采样操作的替换。

这里我使用HWD下采样来代替maxpool操作

5. 替换上采样操作

最后咱们来看上采样操作的替换，例子中的上采样模块只需要传入一个参数，和下采样操作类似，我这里就不过多赘述了

这里我使用Dysample来代替上采样操作